软件学报 ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*,2017,28(2):216−233 [doi: 10.13328/j.cnki.jos.004950]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

# 描述逻辑εL 的二阶线性推理机制*

王 驹[1], 陈光喜[1], 余 泉[2]

[1](广西可信软件重点实验室(桂林电子科技大学),广西 桂林 541004)

[2](黔南民族师范学院 数学系,贵州 都匀 558000)

通讯作者: 陈光喜, E-mail: chgx@guet.edu.cn

**摘 要**: 基于描述逻辑的本体的保守扩充理论、模块抽取理论、通用模块构建理论及其相关算法是本体工程中本体构建、本体融合及重构的核心理论与工具.国际上该领域已有 Lutz 等人使用形式构模方法证明了 ALC 的保守扩充判定算法复杂度是二阶时间指数的,而轻量级的系统εL 的算法复杂度是一阶时间指数的.但当前文献中的形式构模方法思路复杂,难以把握,几乎不能在实用的工程层面上实现.提出一种面向轻量级的描述逻辑系统家族(DL-Lite family)的统一的二阶线性推理机制,并给出该推理机制的完备性证明.该方法直观,思路清晰,从而在工程中容易实现.同时,该方法对εL,$FL_0$,$FL_\varepsilon$,vL 等 DL-Lite 家族的所有系统都有效.在该线序推理系统下,可以根据"空间换时间"的原则,设计和实现关于保守扩充判定的图推理机制,其复杂性(相对于空间的大小)是多项式的.

**关键词**: 描述逻辑;保守扩充;模块本体抽取;二阶线性推理机制;DL-Lite 家族
**中图法分类号**: TP181

中文引用格式: 王驹,陈光喜,余泉.描述逻辑 εL 的二阶线性推理机制.软件学报,2017,28(2):216−233. http://www.jos.org.cn/1000-9825/4950.htm

英文引用格式: Wang J, Chen GX, Yu Q. Second-Order linear reasoning mechanisms for description logic εL. Ruan Jian Xue Bao/Journal of Software, 2017,28(2):216−233 (in Chinese). http://www.jos.org.cn/1000-9825/4950.htm

## Second-Order Linear Reasoning Mechanisms for Description Logic εL

WANG Ju[1], CHEN Guang-Xi[1], YU Quan[2]

[1](Guangxi Key Laboratory of Trusted Software (Guilin University of Electronic Technology), Guilin 541004, China)

[2](Department of Mathematics, Normal College of Minorities of South Guizhou, Duyun 558000, China)

**Abstract**: In the framework of description logics, the theories and their related algorithms of conservative extensions, modularity and module extraction are the core notions and vital tools in engineering semantic Web construction, ontology construction, ontology merging and reuse. Among other important contributions in this area, Lutz, *et al*. have shown that the conservative extension problem for ALC is decidable but its complexity is 2ExpTime-complete while the complexity of the deciding algorithm for light-weight εL is 1ExpTime-complete. Their deciding algorithms are basically depends on tableau algorithm which is substantially a reasoning mechanism in first-order predicate logic. Although, theoretically speaking, those results and algorithms are significant and valuable, both existing theories and methods appear to be complicated, difficult to understand, and hard to implement by engineers working on the semantic Web and ontology construction. This paper will not discuss and analyze the current theories and their algorithms. Instead, it independently

* 基金项目: 国家自然科学基金(61463044, 61363030); 广西自然科学基金(2013GXNSFAA019330); 广西可信软件重点实验室开放基金; 桂林电子科技大学计算机软件创新团队基金(kx201419); 广东省数学教育工程技术研究中心开放基金

Foundation item: Natural Science Foundation of China (61463044, 61363030); Natural Science Foundation of Guangxi Province of China (2013GXNSFAA019330), Open Foundation of Guangxi Key Laboratory of Trusted Software; Foundation of Software Innovative Teamof Guilin University of Electronic Technology (kx201419); Foundation of Guangdong Province Engineering Technology Research Center for Mathematical Educational Software

收稿时间: 2015-08-20; 修改时间: 2015-10-17; 采用时间: 2015-12-05

proposes a second-order linear reasoning mechanism for all members of DL-Lite family. A proof of the completeness of the second-order deduction system is provided. The proposed mechanism is intuitive, easy to manipulate, and much easier to implement in the engineering sector. It is uniformly applicable for members of DL-Lite family such as εL, $FL_0$, $FL_\varepsilon$, and vL. Most of all, this system is consistent with graph deduction that facilitates design and construction of the necessary graphs by using space to exchange with time cost. As a result, the time complexity is reduced significantly such that if the space and deduction graphs are sophisticatedly equipped, the deciding time can be reduced topolynomial.

**Key words**: description logic; conservative extension; ontology module extracting; second-order linear reasoning mechanism; dl-lite family

语义网络及本体构建的理论是理论计算机科学最重要的研究领域之一,描述逻辑是该领域的高层设计的通用语言和体系框架.其中最重要的课题,也是目前的热点,涉及本体的保守扩充、本体的模块化、本体模块抽取、本体重用和重建、本体整合和融合.关于本体保守扩充的判定理论和算法,是上述其他课题的核心指导思想和关键技术.

2007 年,Ghilardi 等人和 Lutz 等人[1,2]提出了保守扩充和模块化等一系列核心概念,用于刻画本体构造中一系列关键问题.他们给出的一个主要结果,是建立了判定 ALC 框架中的本体保守扩充,并证明其复杂度是 2ExpTime- complete.

2009 年,Lutz 等人[3]给出了关于轻量级的描述逻辑系统 εL 的保守扩充判定算法,其复杂度是 1ExpTime-complete 的.他们采用了形式构模的方法,是 Tableau 算法的推广.

国内在本体构建、本体排列的工程层面上,各团队都做了很多有意义的工作,形势喜人.限于篇幅,在此不一一介绍.这里仅简单介绍与本文的保守扩充理论及应用课题密切相关的最新进展:李璞[4]研究了本体模块化重用方法;申宇铭等人和聂登国等人[5,6]解决了 $FL_0$,vL 等轻量级系统的保守扩充判定问题,同时深入讨论了 εL 系统保守扩充算法的复杂性.他们都取得了很好的结果.

本文的主要工作包括以下 3 个方面:(1) 提出了 εL 系统的二阶线性推理机制;(2) 该推理机制适用于所有的 DL-Lite 家族;(3) 使用“以空间换时间”的原则,兼容了图推理,可将指数复杂度$|T_1 \cup T_2|$降低到 $k$, $k$ 是 $T_2$ 中公理的个数(显然, $k<|T_2|$).具体来说,所有必须测试的公理都可以在一个规模为 $2^k$ 的图中解读出来.

## 1 保守扩充

### 1.1 本体的保守扩充、模块抽取、通用模块

在描述逻辑中,本体知识库 *KB* 由 *Tbox*, *Abox* 及解释 *J* 组成[7].在讨论保守扩充和模块抽取时,我们认定一个 *Tbox* 就是一个本体.通常用符号 $T_1, T_2, \ldots$ 来表示不同的本体.若 $T$ 是一个本体,以 $Sig(T)$ 表示在 $T$ 中出现的所有非逻辑符号的集合.例如,若 $T \equiv \{\exists R_1.A \subseteq \exists R_2.B\}$, 则 $Sig(T) = \{R_1, R_2, A, B\}$ ,其中 $A, B, R_1, R_2$ 分别是 $T$ 的原子概念和原子角色.

考虑家庭关系本体 $T_1 : Sig(T) = \{A, B, C, R_1, Person(x); Orphan(x)\}$.

其中, $A = \{\text{Tom}\}; B = \{\text{John}\}; C = \{\text{Tony}\}; R_1 = \{x \text{ is parent of } y\}$, Tom 是 John 的亲生父亲.那么,

$$T_1 \equiv \{A \sqsubseteq \exists R_1(x,y).Person; B \sqsubseteq \exists R_1^-.Person; C \sqsubseteq Orphan\},$$

同时又有 $T_2 : sig(T_2) \equiv \{C; A; R_2 \equiv \{x \text{ Adopted by } y\}\}; T_2 \equiv \{C \sqsubseteq \exists R_2.A; C \sqsubseteq \exists R_1^-.A\}$.

在建立家庭档案时,将 $T_1$ 和 $T_2$ 合并起来就是朴素意义下的本体重构.

在 $T_1$ 中,术语 $C \sqsubseteq \exists R^-(x,y).Person$ 不是 $T_1$ 的逻辑结论,且 $T_1 \cup T_2$ 是不协调的,从而必须去掉术语 $C \sqsubseteq Oraphan$. 这就是朴素意义下的本体融合.

注意到 $\{T_1 \cup T_2\} - \{C \sqsubseteq Orphan\}$ 是协调的,但此时有了一个新的术语 $C \sqsubseteq \exists R^-(x,y).Person$ ,其非逻辑符号在 $Sig(T_1)$ 中.这就使得原来的家庭关系 $T_1$ 发生了变化.按照朴素的看法,扩充后的本体就不是原来的本体的保守扩充.

在很多特殊领域中往往需要将多个小的本体组合成一个新的本体,并希望基本本体的含义不产生变化.这是本体知识工程中一个极其重要的指标,由此产生了一个理论上重要的关键概念.

**定义 1(保守扩充)**[1]. 令 $T_1, T_2$ 是两个不同的本体. $Sig(T_2)-sig(T_1)\neq\varnothing, Sig(T_2)\cap Sig(T_1)\neq\varnothing$. 称 $T_1\cup T_2$ 是 $T_1$ 的保守扩充,如果对所有的形如 $A\sqsubseteq B$ 的包含式,有 $T_1\vdash A\sqsubseteq B$ 当且仅当 $T_1\cup T_2\vdash A\sqsubseteq B$ (注意此处要求 $Sig(A)\cup Sig(B)\subseteq Sig(T_1)$).

例 1:考虑下面的两个本体:

$$T_1\equiv\{A,B,r_1,r_2,A\sqsubseteq\exists r_1.A\}, T_2\equiv\{A,B,r_1,r_2,X_1,...,X_9; A\sqsubseteq X_1, X_1\sqsubseteq X_2,...,X_9\sqsubseteq B\}.$$

可以验证:

$$T_1\nvdash A\sqsubseteq B.$$

构造一个简单的模型如下:

$$\{A=\{a\}, B=\{b\}, r_1=\{\langle a,a\rangle\}, r_2=\{\langle b,b\rangle\}, a\neq b\}.$$

显然,此模型中 $A=\exists r_1.A$ 成立,但 $A=B$ 不成立.但 $T_1\cup T_2$ 是协调的.

模型 $\{A=\{a\}, B=\{a\}, r_1=\{\langle a,a\rangle\}, r_2=\{\langle a,a\rangle\}, X_i=\{a\}\}$ 满足它的所有包含式.

$A\sqsubseteq B$ 是 $T_1\cup T_2$ 的逻辑结论,但不是 $T_1$ 的逻辑结论.所以, $T_1\cup T_2$ 不是 $T_1$ 的保守扩充.

例 2:容易看出,以下的扩充是保守的:

$$T_1\equiv\{A,B,r_1,r_2,A\sqsubseteq\exists r_1.A; A\sqsubseteq B\}.$$

$T_2$ 同例 1.

在本体的构建中,相对于保守扩充,它的逆向工程也是极具重要性的基本操作.给定一个集成本体 $T$,它是由若干个小的本体所构成,常常需要从中找到一些 $T$ 的子集合 $T'\subsetneq T$. 如果从 $T$ 中删去了 $T'$, $T$ 就失去了它应该具有的某些关键属性,而再也不是原来的本体了.描述逻辑对这逆向的工程操作给出了一个很强的、严格的定义.

**定义 2(不可分离)**[2,8]. 令 $T$ 是一个本体(它是若干条包含式的集合). $T'\subset T, Sig(T')$ 是 $T'$ 的非逻辑符号集.称 $T'$ 与 $T$ 是不可分离的(inseparable),若对 $T'$ 的任意包含式 $A\sqsubseteq B, T'\vdash A\sqsubseteq B$ 当且仅当 $T\vdash A\sqsubseteq B$. 也称 $T'$ 为 $T$ 的一个不可分模块(module).

这是狭义的不可分离性的概念.

显然,在定义 2 中, $T$ 一定是 $T'$ 的一个保守扩充.逆向地,在 $T$ 中找出 1 个或多个互不相交的不可分离模块,也是当前本体理论研究中的一个热门课题.对以上的两个热门课题,关键问题就是,任给 $T_1,T_2$,如何找到 $T_1\cup T_2$ 是否为 $T_2$ 的保守扩充的验证算法.

对于本体知识库的维护工程师来说,保守扩充和模块化是重要的.但对本体知识库的构建工程师来说,更期望有一些通用模块:这些通用模块有内部语言,也有外部语言.外部语言只设置在外部接口上(例如做成 Jena 模块的形式).这样,人们就可以用简单的集成办法,如插入、抽出等来构造新的本体.在这个前沿热点课题上,Grau 等人、Lutz 等人和 Wang 等人[9−12]做了大量的工作.

SHIOQ 的语言可表示为 $S=R\uplus C\uplus I$. 这里, $\uplus$ 表示两个不相交集合的并集; $R,C,I$ 分别是角色集合、概念集合、枚举命名集合.该系统的公式构成按惯例.

**定义 3(通用模块类)**. $\mathcal{M}$ 被称为一个通用模块类,如果它的每个元素都是一个 Tbox $T$, 而每个 $T$ 的非逻辑符号都在 $\mathcal{M}$ 划定的内部符号集内, $Sig(T)=Loc(T)\uplus Ext(T)$. 内部语言记为 $Loc(T)$, 外部语言记为 $Ext(T)$. $\mathcal{M}$ 满足:

(1) 对每一个 $T\in\mathcal{M}$, 对任一 $T'$, 只要 $Sig(T')\cap Loc(T)=\varnothing, T\cup T'$ 就是 $T'$ 的保守扩充.

(2) 若 $T_1,T_2\in\mathcal{M}$, 则 $T=T_1\cup T_2\in\mathcal{M}$, 同时 $Loc(T)=Loc(T_1)\cup Loc(T_2)$.

注意到与模块 $T$ 可接口的 $T'$ 不能使用 $T$ 的内部语言,但可以使用 $T$ 的外部语言,两者必须通过 $Ext(T)$ 才能有机地交互、整合起来.为了保证保守扩充这一重要特性,必须对 $T$ 使用 $Ext(T)$ 做严格的限制.记 $E=Ext(T), E\subset S$, 以下归纳定义两组概念(DL公式)的集合.

$$C_E^+ ::= A\,|\,(\neg C^-)\,|\,(C\sqcap C^+)\,|\,(\exists R.C)\,|\,(\exists R.C^+)\,|\,(\geqslant nR.C)\,|\,(\geqslant nR.C^+),$$

$$C_E^- ::= (\neg C^+) \mid (C_1^- \sqcap C_2^-).$$

其中,$A$ 是 $Loc(T)$ 中的原子概念符; $R^+ \in Loc(T)$ ;$C$ 可以含内、外符号; $C^+ \in C_E^+; C_1^-, C_2^- \in C_E^-$.

一个 SHIOQ-TBox 有局部保护性当仅当它的每一个公理的非逻辑符号或者都在 $Loc(T)$ 中,或者是形如 $C^+ \sqsubseteq C, C \sqsubseteq C^-$ 的公式.

**命题 1**[9]. 若 $T$ 具局部保护性,则它一定是通用模块.

近几年关于通用模块的合法性与饱和性有了大量的研究,文献相当多.由此也可以看到以上 3 个方面上层理论研究的重要性.对于局部保护性,可深入研究的问题仍然很多.注意到 $C_E^+, C_E^-$ 的设置,如果我们把它看成一个保护层,它的技术十分复杂,也十分脆弱.人们自然会问:

(1) 这已经是最饱和了的吗?

(2) 是否还有其他更好的设置?

(3) 显然,该设置极大地限制了我们在集成本体中使用语言的表达能力,那么我们能否在具体的情况下放松限制,允许语言更大自由的使用,而仍然保证扩充的保守性质呢?

而这一切可能的深入研究课题的核心技术仍然是保守扩充的验证算法.

迄今为止唯一使用的保守扩充验证算法仍是由 Ghilardi 等人[1]针对 ALC 系统建立的.该方法的复杂度是 2ExpTime-complete.另外,Ghilardi 等人和 Lutz 等人[1−3]在轻量级的 εL 上设计的验证算法的复杂度为 1ExpTime-complete.上述两种验证算法如出一辙,都是形式建模方法(类似于 Tableau 算法).本文的目的不是深入分析这些文献中的缺陷,而是构建一个二阶的、完备的偏序推理机制,保持逻辑完整且具有很强的可操作性.我们的方法适用于几乎所有的 DL-Lite 系统,在很多情况下,具有简单的计算复杂度,易于实现.

### 1.2 DL-Lite家族

考虑描述逻辑 TBox 的语言,其字符包括:

(1) 原子概念符的有穷集合: $B_1, B_2, \ldots, B_n, T, \perp$;

(2) 原子角色符的有穷集合: $r_1, r_2, \ldots, r_m$;

(3) 变元符号集合 $x_1, x_2, \ldots, x_k, \ldots$ 和个体符号 $a_1, a_2, \ldots$ 的集合,它们可以是无穷的;

(4) 量词 $\forall, \exists$;

(5) 逻辑运算符号 $\sqcap, \sqcup, \neg$.

使用到的概念及角色构造子有: $A \sqcap B; A \sqcup B; \neg C; \exists r.C; \forall r.C; R \leqslant n; R \geqslant n; \exists_{\leqslant n} r.C; \forall_{\leqslant n} r.C; \exists_{\geqslant n} r.C; \forall_{\geqslant n} r.C; S_1 \circ S_2$; $S^-$; 以及枚举构造子 $o \equiv \{a_1, a_2, \ldots, a_k\}$; 其中, $a_i$ 是个体符号,通常个体符号是为 ABoX 而设的.

在所有的构造子中, $\neg C$ 是导致复杂度爆发性增加的主要因素.按照构造子的不同,DL 家族大致可分重量级和轻量级两类,比如,ALC,AlCN,SHIOQ 属于重量级;εL,$FL_0$,$FL_\varepsilon$ 属于轻量级.

以下是一些我们感兴趣的轻量级的系统,通常称它们属于 DL-Lite 家族.

εL: $C, D ::= B \mid C \sqcap D \mid \exists r.C \mid \top \mid \perp$.

$FL_0$: $C, D ::= B \mid C \sqcap D \mid \forall r.C \mid \top$.

$FL_\varepsilon$: $C, D ::= B \mid C \sqcap D \mid \exists r.C \mid \forall r.C \mid \top \mid \perp$.

νL: $C, D = B \mid C \sqcup D \mid \exists r.C \mid \top \mid \perp$.

根据不同的特定领域,人们需要不同的构造子来增强系统的表达力.构造子不同的组合还能建立其他 DL-Lite 系统,如:

εLQ: $C, D ::= B \mid C \sqcap D \mid \exists_{\leqslant k} R.C \mid \top \mid \perp$.

εLN: $C, D ::= B \mid C \sqcap D \mid \exists_{\leqslant k} R \mid \exists r.C \mid \top \mid \backslash \perp$.

$\varepsilon L_S^-$: $C, D, R ::= B \mid C \sqcap D \mid R \circ S \mid R^- \mid \exists r.C \mid \top \mid \perp$.

轻量级的 DL 家族系统的优点在于:

(1) 它们的表达力已经远远超过了形式概念分析中的属性符集合.数学符号集合中属性符只是语义记号,符

号之间没有任何逻辑关系,不能表达“存在某个个体具有某种属性;或者所有的个体都具有某种属性”这样的抽象知识.

(2) 它们可以由原子概念出发,构造更多的、在逻辑上有相互联系的概念.

(3) 可以对知识进行真正意义下的推理,而不是简单地进行属性化简和抽取.

由于这些优点,人们已经在数据库的基础上开始使用 DL-Lite 的语言构建本体库,在其中引入推理机制,提高本体库的知识维护和更新的工作效率[12,13].

至于重量级的 DL 系统,特别是 SHIOQ 等,主要面向更高层次的语义网络的构建、交互、访问、搜索等.

## 2 DL-Lite 家族的二阶线性推理机制

通常我们以 FOL 记一阶谓词演算的形式系统(first order logic).Lutz 等人在文献[8,14]中使用典范模型的形式构模方法分别给出判断保守扩充,或者判断 TBox 中子模块是否不可分离的算法,并证明算法的复杂度是 1ExpTime-Complete 的.他们的形式建模方法是在 FOL 中进行的,是一阶的算法.

注意到在很多领域中,二阶的推演往往比一阶的推演更加有力、有效和简单.比如,给出实数的公理系统 $R$,虽然它有可数无穷多个常项符号,也不能证明断言 $\exists x(x^2=2)$ (因为 $\mathcal{Q}$ 是 $R$ 的一个模型,该断言在其中为假.)但如果我们使用实数的类似戴德金分割性质这样的二阶性质,完全可以判定该断言在 $R$ 中为真.证明很常规:函数 $f(x)=x^2-2$ 当 $x=1$ 时取负值,当 $x=2$ 时取正值,该函数曲线是连续的,所以一定与 $x$ 轴相交,从而 $f(x)=0$ 有解.

考虑到 TBox 本体的保守扩充问题,其中的公理及推理,都是关于概念与子概念的关系的断言,也就是子集合之间的包含关系.这是二阶关系.上述事实激发了作者原创性的想法:为保守扩充问题的判定构造一个二阶的推理机制.

本文选择经典的 εL 系统作为平台介绍我们的推理机制构建.至于 DL-Lite 家族的其他任一成员,亦可类似地导入相应的推理规则.

### 2.1 εL的二阶线性推理机制

系统 $\varepsilon L ::= C,D \hookrightarrow B \mid C\sqcap D \mid \exists R.C \mid \top \mid \bot$, 其中 $B$ 是原子概念, $R$ 是原子角色.

**公理**:

(1) $A\sqsubseteq A; \bot\sqsubseteq A; A\sqsubseteq T; A\sqcap P\sqsubseteq A.$

(2) $T_1\equiv\{X_1\sqsubseteq Y_1;\ldots;X_m\sqsubseteq Y_m\}$.

(3) $T_2\equiv\{Z_1\sqsubseteq W_1;\ldots;Z_n\sqsubseteq W_n\}$

**推理规则**:

$\to$-Rule: $\dfrac{A\sqsubseteq B; B\sqsubseteq C}{A\sqsubseteq C}$.

$\sqcap^-$-Rule: $\dfrac{A\sqsubseteq (B\sqcap C)}{A\sqsubseteq C}$.

$\exists^+$-Rule: $\dfrac{A\sqsubseteq B}{\exists R.A\sqsubseteq \exists R.B}$.

$\sqcap^+$-Rule: $\dfrac{A_1\sqsubseteq B_1; A_2\sqsubseteq B_2}{(A_1\sqcap A_2)\sqsubseteq (B_1\sqcap B_2)}$.

**定义 4(框架)**. 令 $A,B$ 是εL 的任意两个不同的公式,其中 $A$ 是 $B$ 的子公式,由于 $A$ 在 $B$ 中可能多次出现,$B$ 可表示为 $B\equiv C[\ldots,A,\ldots,A\ldots]$.从中删除部分或所有的 $A$ 后剩余的部分记为 $C[\ ]$,称其为由 $A,B$ 生成的框架.如果 $A\equiv B$,则生成的框架称为空框架.

例 3:令 $B$ 是原子概念; $r$ 是原子角色. $A\equiv\exists r.B, D\equiv C\sqcap\exists r.B$,则 $A$ 是 $D$ 的子公式,由 $A,D$ 产生的框架既是 $C\sqcap[\ ]$; 若 $D\equiv\exists s\exists r.B$,则由 $A,D$ 生成的框架为 $\exists s[\ ]$;更有:若 $D\equiv\exists s((\exists r.B\sqcap C)\sqcap(\exists t\exists r.B))$,则由 $A,D$ 产生的框架有: $\exists s(([\ ]\sqcap C)\sqcap(\exists t\exists r.B)), \exists s((\exists r.B\sqcap C)\sqcap(\exists t[\ ]))$ 以及 $\exists s(([\ ]\sqcap C)\sqcap(\exists t[\ ]))$.

以下的 4 个规则是上述规则的自然的逻辑推论,在本文的论证中使用更方便(式(2)中的 $C[\ ]$ 是任一框架,在式(3)、式(4)中,$A$ 是 $B$ 的子公式,$C[\ ]$ 是 $A,B$ 生成的框架):

$$\frac{A \sqsubseteq B}{C \sqcap A \sqsubseteq B}, \tag{1}$$

$$\frac{A \sqsubseteq B}{C[A] \sqsubseteq C[B]}, \tag{2}$$

$$\frac{A \sqsubseteq D; E \sqsubseteq B}{E \sqsubseteq B \sqsubseteq C[D]}, \tag{3}$$

$$\frac{D \sqsubseteq A; B \sqsubseteq E}{C[D] \sqsubseteq B \sqsubseteq E}. \tag{4}$$

这 4 个规则分别称为 $\sqcap_+$ -rule; frame-rule; up-rule; down-rule.

以 SL 表示上述推理系统.

**定义 5** (**$SL \vdash A \sqsubseteq B$**). 令 $A,B$ 分别是εL 的任意两个公式(概念),我们称 $SL \vdash A \sqsubseteq B$, 如果存在序列 $A = X_1 \sqsubseteq X_2 \sqsubseteq ... \sqsubseteq X_k = B$, 满足:

(1) $X_1 \sqsubseteq X_2$ 是公理或者是由除 $\to$ -rule 外的某些规则作用到某些公理的逻辑结果.

(2) 当 $i > 1$ 时,每一 $X_i \sqsubseteq X_{i+1}$ 都是某一规则的逻辑结果,而该规则的前提是公理或者是已证包含式,其中 $X_1 \sqsubseteq X_2 ... \sqsubseteq X_i$ 都是已证公式.

注:任意包含式 $A \sqsubseteq B$ 在 FOL 中都可理解为全称断言: $\forall x(A(x) \to B(x))$. 容易验证,以上罗列的规则在一阶谓词逻辑中是有效的.所以,下面的引理是自明的.

**引理 1**. $SL + T_1 + T_2 \vdash A \sqsubseteq B$, 则 $FOL + T_1 + T_2 \vdash A \sqsubseteq B$.

### 2.2 SL推理系统的完备性

以 $\mathcal{F}$ 记系统εL 中所有由 $Sig(T_1 \cup T_2)$ 的符号产生的公式(概念)组成的集合.而集合 $\mathcal{F}_1$ 是 $Sig(T_1)$ 产生的公式的全体.任给 $A,B \in \mathcal{F}$, 若在不使用 $\to$- 规则的前提下能得到 $SL \vdash A \sqsubseteq B$ (记为 $A \leqslant B$ ),那么 $\langle \mathcal{F}; \leqslant \rangle$ 构成一个(无穷)半序集合.显然, $\langle \mathcal{F}; \leqslant \rangle$ 也是半序集合.半序集合的路径含义自明,但路径必须是有穷个公式组成的由左至右的链.若 $A,B$ 在 $\langle \mathcal{F}; \leqslant \rangle$ 中是联通的,记为 $A \rightsquigarrow B$. 显然,它也意味着 $A \sqsubseteq B$. 注意,我们将 $\rightsquigarrow$ 理解为向右的有向链接,它是有穷个 $\leqslant$ 的组合.

另外,注意到系统 $\langle \mathcal{F}; \leqslant \rangle$ 和 $\langle \mathcal{F}_1; \leqslant \rangle$ 不一定具有类似于格那样的结构,在其中有可能出现循环的现象.这是由于 $A \rightsquigarrow B$ 和 $B \rightsquigarrow A$ 有可能同时成立.

容易验证以下引理.

**引理 2**. 若 SL 是完备的,则 $T_1 \cup T_2$ 不是 $T_1$ 的保守扩充当且仅当存在 $A,B \in \mathcal{F}_1$, 在 $\langle \mathcal{F}; \leqslant \rangle$ 中,关系 $A \rightsquigarrow B$ 成立,但在 $\langle \mathcal{F}; \leqslant \rangle$ 中不成立.

在无穷半序结构 $\langle \mathcal{F}; \leqslant \rangle$ 中,将关系 $\leqslant$ 简化为有向连接边 $\to$ ,则该结构是一个有向半序结构,约定若 $A \rightsquigarrow B$, 则 $A$ 处于该有向路径的左部, $B$ 则位于右部.

下面考虑完备性(在以下的论证中,$SL$ 仅代表推理规则的集合, $T$ 是任意的 TBox).

**定理 1**(**SL 系统的完备性**). 令 $T$ 是εL 语言框架中的一个本体,它由形如 $A \sqsubseteq B$ 的有穷个包含式组成.对任意的εL 语言中的包含式 $X \sqsubseteq Y$, 有: $FOL + T \vdash X \sqsubseteq Y$ 当且仅当 $SL + T \vdash X \sqsubseteq Y$.

要证 SL 的完备性,由引理 1,只要证明它的逆即可,即,若 $SL + T \nvdash A \sqsubseteq B$ 则必有 $FOL + T \nvdash A \sqsubseteq B$.

证明的主思路是构建 $SL + T$ 的一个形式模型 $J$, 使得对任意的包含式 $A \sqsubseteq B$, $J \vdash A \sqsubseteq B$ 当且仅当 $SL + T \vdash A \sqsubseteq B$. 那么,若 $SL + T \nvdash A \sqsubseteq B$, 则该包含式在 $J$ 中不成立,由于 $J$ 是 $T$ 的模型,它当然也是 $FOL + T$ 的模型.由完全性定理,当然有 $FOL + T \nvdash A \sqsubseteq B$.

注意到在所有的εL 公式中,个体变量只需要有限个.考虑基本概念符 $B$ 和角色符 $R$:任意的解释域$\Delta$和任意的解释 $J$ 给出后,公式 $\exists R(x,y).B$ 和 $\exists R(z,w).B$ 的含义相同,按照定义,总有 $[\exists R(x,y).B]^J = [\exists R(z,w).B]^J$. 在文献中,一

般情形下,变元符号都不写出来.我们给 $\mathcal{F}$ 的所有元素都分配一个唯一的、两两不等的正整数,如下所示.

(1) 字符表:$\exists,(,),\sqcap,B_1,B_2,...,B_m,R_1,R_2,...,R_n$(此处略去了符号 $\cdot$,它不影响对公式的辨认).

(2) 将 $1,2,...,m+n+4$ 按顺序分别赋予每一个符号,以这些整数作为各个符号的分配数.

(3) 令 $1<p_1=2,p_1=3,...,p_k,...$ 是大于 1 的质数的无穷序列.

(4) 每一个原子公式,$B_i\,|\,1\leqslant i\leqslant m$, 赋予它整数 $p_1^{i+4}$.

(5) 对任意公式,若它的符号个数是 $k$,则赋予其整数 $p_1^{i_1}p_2^{i_2}...p_k^{i_k}$, 其中, $i_j$ 是该公式第 $j$ 个字符的分配数.例如,公式 $A\equiv\exists R_1.((\exists R_3.B_1)\sqcap(\exists R_2.B_3))$, 有 15 个字符.按以上配置,赋予它的整数是 $2^1\cdot3^{m+5}\cdot5^2\cdot...\cdot p_{15}^2$.

**引理 3**. 任给 $F$ 中两个不同的公式 $A,B$, 分配给它们的整数是不等的.

证明:以 $u(A)$ 记的分配数 $A$.根据配置,若 $A$ 的长度不等于 $B$ 的长度,不妨设 $|A|>|B|$.于是, $A$ 的分配数的质数因子分解中应有至少 1 个因子不在 $u(B)$ 的分解中出现,所以 $u(A)\neq u(B)$. 如果 $|A|>|B|$, 因为两公式是不同的,则在某一位置 $i$ 上, $A$ 的字符与 $B$ 的字符不同,字符的分配数应是 $k_1\neq k_2$, 那么, $p_i^{k_1}\neq p_i^{k_2}$, 从而 $u(A)\neq u(B)$. □

令 $\Delta=\{u(A)\,|\,A\in F\}$, 由此我们将要合理地把 $\Delta$ 中的元素逐步放入各个公式中,然后将其扩充为 $T_1+T_2$ 的一个模型 $J$. 以下 $J^-$ 是 $J$ 的预置, $A^{J^-}$ 表示对公式 $A$ 或符号 $A$ 的预置.

- 对每一个原子概念 $B_i$, 将 $u(B_i)$ 直接放入 $B_i^{J^-}$ 中.
- 对每一原子概念 $B_i$ 和角色符 $R^j$, 将序对 $\langle u(\exists R^j.B_i),u(B_i)\rangle$ 放入 $R_i^{J^-}$ 中.从而 $u(\exists R^j.B_i)$ 成为 $(\exists R^j.B_i)^{J^-}$ 的一个元素.
- 归纳地,若 $A\equiv D\sqcap C$, 则将 $u(A)$ 同时放入 $D^{J^-}$ 和 $C^{J^-}$ 中.
- 同样归纳地,若 $A\equiv\exists R^j.C$,则将序对 $\langle u(A),u(C)\rangle$ 放入 $R_i^{J^-}$.

在描述逻辑语义的定义下,我们有:

**引理 4**. 对每一个 $A\in F$, $A^{J^-}$ 含有自己的分配数 $\mu(A)$.同时含有所有型为 $u(A\sqcap B)$ (对所有 $B\in F$)的公式的分配数.

注意, $A\sqcap A$ 看成一个公式 $A$; 任意 $A,B,A\sqcap B\sqsubseteq A$ 是集合论的一条公理(也可以由规则 $\sqcap^-$ 通过公理 $A\sqcap B\sqsubseteq A\sqcap B$ 得到).更详细的内容可以由归纳法证明,此处从略.将公式集合及分配数集合分别排列成可数序列:

$$A_1,A_2,...,A_n,...$$

$$u(A_1),u(A_2),...,u(A_n),...$$

**定义 6(模型 $J$ 的构成)**. 对每一个 $A\in F$ 建立集合 $A^J: u(A_j)\in A^J$ 当且仅当 $SL+T_1+T_2\vdash A_j\sqsubseteq A$(对每一角色符 $R_i$ 也必须捆绑式地定义 $R_i^J$, 见下文).在将 $u(A_j)$ 放入 $A^J$ 时,依照以下的规则.

- 如果 $A$ 是原子概念 $B_i$, 则将 $u(A_j)$ 直接放入 $B_i^J$ 中.
- 如果 $A$ 是 $\exists R^j.B_i$, 则将序对 $\langle u(A_j),u(C)\rangle$ 放入 $R_i^J$.
- 归纳地,若 $A\equiv D\sqcap C$, 则将 $u(A_j)$ 同时放入 $D^J$ 和 $C^J$ 中.
- 同样归纳地,若 $A\equiv\exists C$, 则将序 $\langle u(A_j),u(C)\rangle$ 放入 $R_i^J$.

注意,当且仅当 $SL+T\vdash A_j\sqsubseteq A$ 时,我们才执行以上的操作.所以,可以归纳地证明: $A^J,R_i^J$ 的定义不仅是合理的,而且更重要的是它符合了传统的描述逻辑语义的规则.从而只要证明 $J\models T_1+T_2$, 则可知它是 $T_1$+$T_2$ 在描述逻辑语义构建意义下的一个模型.

**引理 5**. $J$ 是 $T$ 的模型.

证明:令 $X\sqsubseteq Y$ 是 $T$ 的任一公理.我们证 $X^J\subseteq Y^J$. 首先, $X\sqsubseteq Y$ 是 $SL+T$ 可证的,按 $J$ 的定义 $u(X)\in Y^J$. 对任意整数 $a\in X^J$, $a$ 必须是某一公式 $A_i$ 的分配数 $u(A_i)$. 由 $J$ 的定义, $a\in X^J$, 而 $a=u(A_i)\in X^J$ 意味着 $SL+T\vdash A_i\sqsubseteq X$. 从而 $SL+T\vdash A_i\sqsubseteq Y$, 由 $J$ 的设置,我们有 $a\in Y^J$. 因而 $X^J\subseteq Y^J$. 从而 $J$ 是 $T_1+T_2$ 的模型. □

注意到以下的事实:

**断言 1**. 对每一公式 $A,A^J$ 所含元素都是某一公式 $B$ 的分配数 $u(B)$, 也相当于 $B\sqsubseteq A$ 是在系统 $SL+T$ 中可证的.

对任意两公式 $X,Y$,若 $SL+T\nvdash X\sqsubseteq Y$,则 $u(X)\in X^J$,同时 $u(X)\notin Y^J$; 又若 $SL+T_1+T_2\nvdash Y\sqsubseteq X$,则 $u(Y)\in Y^J$. 同时,$u(Y)\notin X^J$.

注意到 $J$ 也是系统 $FOL+T_1+T_2$ 的一个模型.其中,$X\sqsubseteq Y;Y\sqsubseteq X$ 都也是不可证的.这就证明了 SL 的完备性:对任何两个公式 $X,Y,SL+T_1+T_2\vdash X\sqsubseteq Y$ 当且仅当 $FOL+T_1+T_2\vdash X\sqsubseteq Y$.

注:在集合论的公理系统保证下,一般来说,任一由若干包含式形成的公理系统都是协调的.以下的模型可以证明以上两个系统的协调性:令 $Sig(T)=\{A,B,C,D\}.\{T\equiv A\sqsubseteq B;B\sqsubseteq C\}$, 这里, $A,B,C,D$ 都是原子概念.定义解释 $J:A^J=\{a\},B^J=\{a,b\},C^J=\{a,b,c\},D=\varnothing$.注意到,在 $J$ 中,FOL 和 SL 都是有效的.而 $SL$+$T\nvdash C\sqsubseteq D$.所以,SL 是协调的.否则,它可以推出任何结论,包括 $C\sqsubseteq D$ .

## 3 SL 的推理模式及其关键技术

要验证 $T_1\bigcup T_2$ 是不是 $T_1$ 的保守扩充,由引理 2,只需要检查形如 $A\to B$ 的包含关系链.假定 $Sig(T_1)$ 的符号都是红色的,$Sig(T_2)-Sig(T_1)$ 的符号是黑色的.如果一个公式含有黑色的非逻辑符号,就看作是灰色的.只需要考察形如 $A\sim B:A\sqsubseteq D_1\sqsubseteq D_2\sqsubseteq...\sqsubseteq D_k\sqsubseteq B$ 的包含链,其中 $A,B$ 是红色的,而其他所有的 $D_i$ 都是灰色的.这是因为,若某一 $D_i$ 是红色的,同时 $A\sqsubseteq B$ 又不是 $T_1$ 的定理(从而 $T_1\bigcup T_2$ 不是 $T_1$ 的保守扩充),那么 $A\to D_i$ 以及 $D_i\to B$ 中一定有一个不是 $T_1$ 的定理.称这样的链为有效链.反过来,如果每一个有效链 $A\to B$ 都能找到一条红色的链连接 $A,B$ ,则扩充一定是保守的.

### 3.1 框架及标准包含链的延长

假定在 $\langle F;\leqslant\rangle$ 中存在一条包含链 $A_0\sqsubseteq A_1\sqsubseteq...\sqsubseteq A_k$,对每一个 $i,0<i\leqslant k,A_{i-1}\sqsubseteq A_i$ 都是 SL 的上述给定的逻辑或非逻辑公理或者是由它们使用 frame-rule 的结果.称这样的链为标准链.约定以后的链都是标准链.

有些包含链可能是无穷长的:如果术语集中有形如 $A\sqsubseteq\exists R_1.[A\bigcap\exists R_2.B]$ 的一条非逻辑公理,按 up-rule 规则,有:

$$A\sqsubseteq\exists R_1.[A\bigcap\exists R_2.B]\sqsubseteq\exists R_1.[\exists R_1.[A\bigcap\exists R_2.B]\bigcap\exists R_2.B]\sqsubseteq...\sqsubseteq...$$

这是在框架 $\exists R_1.[()\bigcap\exists R_2.B]$ 中无穷次迭代上述公理的结果.但这样的链是无效的,无需检查.因为或者它全是红的,或者从第 2 件起,全是灰色的.

还有另一种产生无穷链的方式: $A\sqsupseteq A\bigcap P_1\sqsupseteq A\bigcap P_1\bigcap P_2\sqsupseteq...\sqsupseteq A\bigcap P_1\bigcap...\bigcap P_k\sqsupseteq...$ 同样地,这样的链也是无效的.

**引理 6**.

(1) 给定的链 $A_0\sqsubseteq A_1\sqsubseteq...\sqsubseteq A_k$ 可向右添加一个公式 $A_{k+1}$, 当且仅当有 SL 的公理 $D_1\sqsubseteq D_2$, 使得 $D_1$ 是 $A_k$ 的子公式(可能有多个出现),在 $A_k$ 中删除 1 个或多个 $D_1$ 的出现,形成一个 $C[\ ]$ 框架,记为 $A_k\equiv C[D_1]$. 令 $A_{k+1}\equiv C[D_2]$. 显然,此时 $A_k\sqsubseteq A_{k+1}$ 成立.

(2) 上述给定的链可向下添加公式 $A_{-1}$ ,当且仅当有 SL 的公理 $D_1\sqsubseteq D_2$, 使得该包含式的后件是 $A_0$ 的子公式,后件与 $A_0$ 产生的框架记为 $C_1[\ ]$ (可能为空),使得 $A_0\equiv C_1[D_2];A_{-1}\equiv C_1[D_1]$.

该引理使得我们可以分析哪些框架在构建链的加长中起了作用.注意到 SL 中的公理集合是有穷集合,定义 $C$=$\{\langle C[\ ],A,B\rangle\}$, 这里, $A$ 是某公理的前件(后件), $B$ 是某公理的后件(前件). $A$ 是 $B$ 的子公式, $C[\ ]$ 是 $A,B$ 产生的框架.

**引理 7**.

(1) 在向右延长包含链时,使用的框架是 $C$ 的元素或者是有穷个元素的复合.

(2) 在向左延长包含链时,如果对 $\bigcap_+$ 的公式约束在术语集合的公式中,则上述结论亦成立.

用归纳法很容易验证该引理.

上述事实给出了有效链的构成的一般形态.但全局的情形远远比这要复杂得多.

例 4:令

$Sig(T_1)=\{r,s,A,B\},T_1\equiv\varnothing$.

$T_2 \equiv$:

- $A \sqsubseteq Y_1 \sqcap Y_2 \sqcap Y_3$;
- $\exists r.(Y_1 \sqcap Y_2) \sqcap \exists s.(Y_1 \sqcap Y_2) \sqsubseteq X_i, 1 \leqslant i \leqslant 3$;
- $X_1 \sqcap X_2 \sqcap X_3 \sqsubseteq B$.

由此看不出有任何链的衔接,也找不出任何有效链.那么,$T_1 \cup T_2$ 是 $T_1$ 的保守扩充吗?事实并非如此.这是因为,由公理 $A \sqsubseteq Y_1 \sqcap Y_2 \sqcap Y_3$,使用 $\sqcap^-$ 规则可以得到 $A \sqsubseteq Y_1 \sqcap Y_2$,此包含式的右件是第 2 个包含式左件的子公式,而且有两次出现,由框架规则,有 $(\exists r.A) \sqcap (\exists s.A) \sqsubseteq \exists r.(Y_1 \sqcap Y_2) \sqcap \exists s.(Y_1 \sqcap Y_2) \sqsubseteq X_i, 1 \leqslant i \leqslant 3$,更有 $(\exists r.A) \sqcap (\exists s.A) \sqsubseteq X_i$,$1 \leqslant i \leqslant 3$,由规则 $\sqcap^+$,有 $(\exists r.A) \sqcap (\exists s.A) \sqsubseteq (X_1 \sqcap X_2 \sqcap X_3)$.

于是,$(\exists r.A) \sqcap (\exists s.A) \sqsubseteq B$.

注意到该包含式的非逻辑符号集等于 $Sig(T_1)$,由定义可知,$T_1 \cup T_2$ 并不是 $T_1$ 的保守扩充.

注:有趣的是,$T_1 \cup T_2$ 证不出 $A \sqsubseteq B$.读者可自行验证.

### 3.2 有效链的形成特征、长度及前后端公式的宽度

设存在有效链 $A \sqsubseteq X_1 \sqsubseteq X_2 \sqsubseteq ... \sqsubseteq X_k \sqsubseteq B$.满足 $Sig(A) \cup Sig(B) \sqsubseteq Sig(T_1), Sig(X_i) \cap (Sig(T_2) - Sig(T_1)) \neq \varnothing; \forall i$;$1 \leqslant i \leqslant k$;并且 $A,B$ 是红色的,每个 $X_i$ 都是杂色的.

为了方便,也用小写希腊字母 $\gamma,\lambda,\sigma$ 来记一般公式.又以 $C \in \mathbb{C}$ 记 $T_1 \cup T_2$ 中的基本框架,以 $\Gamma, \Gamma_i$ 记复合框架.

研究一个逻辑推理系统的推理形态,最重要的是考虑其全局复杂性:因为形式推理往往是极其发散的、多路径的(例如,数学定理的证明过程就是搜索式的).因而要刻画有效链的构成特征,切入点就是先要搜索到那些关键的构成包含链的属性,也即新的概念.这些新的概念就像摸着石头过河的那些石头一样扎实、可靠,是达到研究目的的不可或缺的基础.

给出一个灰色公式 $A$,其严格的数学含义是:$Sig(A) \cap Sig(T_2) \neq \varnothing$.

**定义 7(核)**. 任给一个灰色公式 $A$,它的核是指 $A$ 的最小公式 $B$,满足:

(1) $Sig(B) \cap Sig(T_2) \neq \varnothing$;

(2) 记 $C[\ ]$ 是由 $B,A$ 产生的框架(满足 $A \equiv C[B]$),必有 $Sig(C[\ ]) \subseteq Sig(T_1)$.

设 $X \sqsubseteq Y \in T_2$,如果 $Sig(X) \subseteq Sig(T_1), Sig(Y) \cap Sig(T_2) \neq \varnothing$,即 $X$ 是红色的,$Y$ 是灰色的,则称其为有效左件.反过来,若 $X$ 是灰色的,$Y$ 是红色的,则称其为有效右件.

容易证明:任一公式的核是唯一的.当然,某些公式的核就是它本身.

$T_2$ 中也可能含有一类公理 $X \sqsubseteq Y$,左、右端的公式都是红色的,是最小的有效链.容易证明,如果 $T_2$ 没有最小有效链,或者所有的最小有效链都是 $T_1$ 的定理,同时 $T_2$ 又没有有效左(右)件,则 $T_1 \cup T_2$ 一定是 $T_1$ 的保守扩充.

另外,$T_2$+$SL$ 可以产生很多的新的定理,它们具有有效左(右)件的色彩特征(例 4 中就有若干具体实例),有必要筛选并除掉那些冗余的有效左、右件,为此引入以下定义.

**定义 8(简单向左生成链)**. 以符号 $A \Rightarrow B$ 表示 $A$ 是 $B$ 的子公式.简单向左生成链指的是在 $T_2$ 中一组公理 $A_i \sqsubseteq B_i, 1 \leqslant i \leqslant k-1$,使得:

(1) $B_i \Rightarrow A_{i+1}$,且 $B_i$ 在 $A_{i+1}$ 中只出现 1 次.

(2) $Sig(C_i[\ ]) \subseteq Sig(T_1), C_i[B_i] \equiv A_{i+1}$,而且除 $A_1$ 是红色的之外,其他公式都是灰色的.

由此可逐步构成如下的向左生成的有效链(请注意框架出现的顺序及其位置).

(1) $C_{k-1}[A_{k-1}] \sqsubseteq C_{k-1}[B_{k-1}] \equiv A_k \sqsubseteq B_k$;

(2) $C_{k-1}[C_{k-2}[A_{k-2}]] \sqsubseteq C_{k-1}[C_{k-2}[B_{k-2}]] \equiv C_{k-1}[A_{k-1}] \sqsubseteq C_{k-1}[B_{k-1}] \equiv A_k \sqsubseteq B_k$;

(3) ...

(4) $C_{k-1}[C_{k-2}[...C_1[A_1]...]] \sqsubseteq C_{k-1}[C_{k-2}[...C_2[C_1[B_1]]...]] \equiv C_{k-1}[C_{k-2}[...C_2[A_2]...]] \sqsubseteq C_{k-1}[A_{k-1}] \sqsubseteq C_{k-1}[B_{k-1}] \equiv A_k \sqsubseteq B_k$.

该链可简写成路径形式(见第 3.3 节的设置): $A_1 | B_1 \xrightarrow{C_1} A_2 | B_2 \xrightarrow{C_2} \cdots \xrightarrow{C_{k-1}} A_k | B_k$.

同样地,可以定义简单向右生成链: $A_i \Rightarrow B_{i-1}: 2 \leqslant i \leqslant k$. 这里所有的框都是红色的.

$$\begin{aligned} A_1 \sqsubseteq B_1 &\equiv C_1[A_2] \sqsubseteq C_1[B_2] \\ &\equiv C_1[C_2[A_3]] \sqsubseteq C_1[C_2[B_3]] \sqsubseteq \ldots \sqsubseteq C_1[C_2[\ldots C_{k-1}[B_{k-1}]\ldots]] \\ &\equiv C_1[C_2[\ldots C_{k-1}[A_k]\ldots]] \sqsubseteq C_1[C_2[\ldots C_{k-1}[B_k]\ldots]]. \end{aligned}$$

进一步地,考虑序列: $A_1 \sqsubseteq B_1 \equiv A_2 \sqsubseteq B_2 \Rightarrow \ldots A_k \sqsubseteq B_k \Leftarrow A_{k+1} \sqsubseteq B_{k+1} \Leftarrow \ldots \Leftarrow A_n \sqsubseteq B_n$.

其中,除 $A_1, B_n$ 是红色的,其他公式都是灰色的,而且子公式关系 $\Leftarrow, \Rightarrow$ 产生的框架都是红色的.因此我们得到一个有效链,它的两端红色公式的长度都远远超过 $|A_1|, |B_n|$.

$$\begin{aligned} C_{k-1}[C_{k-2}[\ldots C_1[A_1]\ldots]] \sqsubseteq C_{k-1}[C_{k-2}[\ldots C_2[C_1[B_1]]\ldots]] &\equiv C_{k-1}[C_{k-2}[\ldots C_2[A_2]\ldots]] \sqsubseteq C_{k-1}[C_{k-2}[\ldots C_2[B_2]\ldots]] \\ &\ldots \\ &\equiv \sqsubseteq C_{k-1}[A_{k-1}] \sqsubseteq C_{k-1}[B_{k-1}] \\ &\equiv A_k \sqsubseteq B_k \\ &\equiv C_k[A_{k+1}] \sqsubseteq C_k[B_{k+1}] \\ &\equiv C_k[C_{k+1}[A_{k+2}]] \sqsubseteq C_k[C_{k+1}[B_{k+2}]] \\ &\ldots \\ &\equiv C_k[C_{k+1}[\ldots C_{n-1}[A_n]\ldots]] \sqsubseteq C_k[C_{k+1}[\ldots C_{n-1}[B_n]\ldots]]. \end{aligned}$$

考虑上述定义中的简单向左生成链.可以发现以下的有价值的性质:

1) 每一向左生成链的最左件都是有效左件.如果每一 $C_i[\ ]$ 都非空,则最左的红色公式的宽度增加.

2) 每一个 $B_i$ 的核必定与 $A_{k+1}$ 的核是相等的,即它们是同一公式.以 $C_0(X)$ 记公式 $X$ 的核.若 $C_0(B_i)$ 是 $C_0(A_{k+1})$ 的真子公式,则 $C_0(A_{k+1}) \equiv C[C_0(B_i)]$ 那么 $C[\ ]$ 必定包含黑色非逻辑符号.否则,要么 $B_i \Rightarrow A_{k+1}$ 不成立,要么 $C_i[\ ]$ 是灰色的,都导致矛盾.

3) 如果有 $1 \leqslant i \neq j \leqslant k$ 使得 $A_i \equiv A_j, B_i \equiv B_j$,则生成过程会产生循环,可以任意扩大有效左件的宽度.

4) 若 $\gamma_1, \gamma_2$ 分别是简单向左(向右)生成链,$\gamma_1$ 的最右公式是 $B_k$,$\gamma_2$ 的最左公式也是 $B_k$,则 $\gamma_1 \sim \gamma_1$ 合成一条有效链,这样的有效链也是简单的.

5) 如果 3)中刻画的情形发生(也可以发生在向右生成链中),则循环会产生任意长的有效链,以及任意宽的红色两端(中间为灰色公式).

对于全局的情形,在上述定义中,假设某些 $B_i$ 在 $A_{i+1}$ 中多次出现,若某些 $C_i[\ ]$ 仍然是灰色的,仍然可以构成有效链.为此,我们将构建一个推理图结构 $\tau$,其中的公式长度不超过 $T_1 \cup T_2$ 中最长公式的长度.相比之下,Lutz 等人的方法要求核查许多的“嵌套深度”的界是 $|T_1 \cup T_2|$ (所有在其中出现的公式长度之和),这样的公式长度更是远远大于 $\tau$ 中公式的长度,而 $\tau$ 中的公式个数仅比 $T_1$+$T_2$ 中的公式个数多一些,但所有的有效链的信息都在其中被刻画出来(通常是以图结构中路径的方式).

### 3.3 推理图 $\tau$ 的设置及其饱和性

以整数 $p$ 记 $T_1, T_2$ 的所有包含式中最长的公式(它应该是某一包含式的左端公式或右端公式)的长度.

#### 3.3.1 推理图 $\tau$ 的构造

记 $\tau =:: \langle \Gamma_1; \Delta; \Gamma_2; \xrightarrow{C} \rangle$.

• $\Delta \subseteq T_2$ 是 $T_2$ 中的所有红色包含式的集合,称为平凡有效链.

• 对 $T_2 - \Delta$ 分别使用 $\sqcap^-, \sqcap^+, \exists^+, \sqcap_+$ 规则,得到新的包含式 $X \sqsubseteq Y$ 满足约束条件: $|X| \leqslant p; |Y| \leqslant p$. 在若干步的操作后,得到一个包含式的集合 $\Gamma_2$,它对上述 4 种规则的操作封闭,且具有如下性质:

(1) $T_2 \subset \Gamma_2$;

(2) $\Gamma_2$ 的每一包含式 $X \sqsubseteq Y$ 或是原来的公理,或是导出的定理,同时 $|X| \leqslant p; |Y| \leqslant p$.

• 对 $T_1$ 同样进行上述的操作,得到包含式集合 $\Gamma_1$.

• 分别将 $\Gamma_1, \Gamma_2$ 按某种方式写成矩阵的形式(矩阵中允许有空的位置 $S_{ij} = \varnothing$; 每一非空 $S_{ij}$ 的是一个形如

$X \sqsubseteq Y$ 的包含式;$\Delta$ 写成 $n\times 1$ 的列向量形式;将各部分整合排列成 $\Gamma_1|\Delta|\Gamma_2$ 的形式).

• 此外,对其中子矩阵 $\Gamma_2$ 部分,要求将其所有有效左件写在第 1 列的位置,而有效右件都写在最右列.分别构成左、右两个竖直向量.其他包含式可按长度或字母表序构成子矩阵.

• $\xrightarrow{C}$ 的设置:

– 以 $\alpha,\beta$ 分别表示红色类型、灰色类型的框架.对 $\Gamma_1;\Gamma_2$ 的包含式 $X \sqsubseteq Y$, 若 $X$ 是 $Y$ 的子公式,则以向右弯曲连线将 $X$ 与 $Y$ 连接,箭头指向 $Y$,同时标注出框架及框架类型,如 $X \xrightarrow{\gamma\cdot C} Y;\gamma\in\{\alpha,\beta\}$. 反之,则用向左弯曲连线连接,形如 $X \xleftarrow{\gamma\cdot C} Y;\gamma\in\{\alpha,\beta\}$ (称为带标记连接边).

– 一般地,令 $A$ 是 $\Gamma_2$ 中某一包含式的左件, $B$ 是另一包含式的右件.若 $A \Rightarrow B$, 则连接为 $A \xrightarrow{\gamma\cdot C} B$; $\gamma\in\{\alpha,\beta\}$. 反之,连接为 $A \xleftarrow{\gamma\cdot C} B;\gamma\in\{\alpha,\beta\}$. 若 $A\equiv B$, 则连接为 $A \leftrightarrow B$.

– 令 $\Sigma$ 是由矩阵 $\Gamma_2$ 的最左列和最右列组成的 $n\times 2$ 矩阵.对子图 $\Gamma_1|\Delta|\Sigma$ 以上述方式完成所有的连接.当然,连接边的标记都是 $\alpha$ 类型的框架.约定:

(1) 若 $A$ 是 $\Gamma_1$ 的某一包含式的左件,而 $B$ 是 $\Delta$ 包含式的右件或者 $\Sigma$ 左列的公式,同时 $A$ 是 $B$ 的子公式,添加链接 $A \xrightarrow{\alpha\cdot C} B$.

(2) 对称地,若 $A$ 是 $\Gamma_1$ 的某一包含式的右件,而 $B$ 是 $\Delta$ 某一公式的右件,或是 $\Sigma$ 右列的一个公式,同时 $A$ 是 $B$ 的子公式,则添加链接 $A \xrightarrow{\alpha\cdot C} B$. 注意,在对有效路径的解读(见第 3.2.2 节)中,我们不对该类链接的 $B$ 做任何的解读,它仅是解读过程中 $A$ 是否能链接到 $B$ 的参考.

例 5: $\Gamma_2=\{A_1\sqsubseteq X_1;A_2\sqsubseteq X_2;X_1\sqsubseteq Y_1;X_2\sqsubseteq Y_2;Y_1\sqcap Y_2\sqsubseteq C\}$, 且 $\{A_1,A_2,C\}\subset Sig(T_1)$, 有:

$$A_1|X_1\leftrightarrow X_1|Y_1\rightarrow Y_1\sqcap Y_2|C,$$
$$A_2|X_2\leftrightarrow X_2|Y_2\rightarrow Y_1\sqcap Y_2|C.$$

它们合成有效链 $A_1\sqcap A_2\sqsubseteq X_1\sqcap X_2\sqsubseteq Y_1\sqcap Y_2\sqsubseteq C$.

**定义 9($\Gamma_2$ 的基本包含式)**. 不使用 $\rightarrow$-rule, 由子系统 $(SL-\{\rightarrow\text{-rule}\}+\Gamma_2)$ 推出的所有包含式称为基本包含式.记基本包含式的链长为 1.

**断言 2**. 任何一个由 $SL+\Gamma_2$ 推出的包含式 $X\sqsubseteq Y$ 都可以由基本包含式构成一个链 $X\equiv\gamma_0\sqsubseteq\sigma_1\sqsubseteq\sigma_2\sqsubseteq\ldots\sqsubseteq\sigma_k\sqsubseteq\gamma_1\equiv Y$, 且 $\{\gamma_0\sqsubseteq Y;\sigma_i\sqsubseteq\sigma_{i+1}|1\leqslant i\leqslant k\}$ 中的成员都是基本包含式.

证明:由定义, $X\sqsubseteq Y$ 是可证的,必定存在一个有穷的证明序列 $\omega_1,\omega_2,\ldots,\omega_n\equiv X\sqsubseteq Y$ (所有的 $\omega_i$ 都是包含式),其中每一个 $\omega_k$ 或者是 $\Gamma_2$ 的公理,或者是按某一规则,以若干 $\omega_k$ 前面的包含式为前提,执行该规则得到的结果.如果这个证明序列没有使用 $\rightarrow$-rule, 则 $X\sqsubseteq Y$ 是基本包含式;如果 $\rightarrow$-rule 使用了 1 次,而且使用后的结果就是 $X\sqsubseteq Y$, 那么其前提必然形如 $X\sqsubseteq Z$; 如果 $Z\sqsubseteq Y$, 那么它们必须都是基本包含式.由此,施归纳法于 $\rightarrow$-rule 的次数,即可完成证明.显然,断言中的有效链的链长为 $k+1$. □

为进一步讨论,需要注意到以下的事实:

(1) 若 $A$ 是 $B$ 的子公式, $A$ 可能在 $B$ 中有多处出现.如果有: $C|A\rightarrow B|D$ 可以同时将 $C$ 替换 $A$ 的每一次出现,也可以一次接一次地替换.

(2) 在一个公式中 $A\sqsubseteq B$ 中,若 $Core(A)$ 定义为最小的灰色的子公式,对于 $B_1,B_2\in Core(A)$,则它们是不相交的,即 $B_1,B_2$ 在 $A$ 中没有公共部分.

令 $A\equiv X_0\sqsubseteq X_1\sqsubseteq X_2\sqsubseteq\ldots\sqsubseteq X_k\sqsubseteq X_{k+1}\equiv B$ 是一个有效链,其每一连接都是基本包含式,深入到每一基本包含式的结构中,可以得到很多信息.事实上, $k$ 个基本包含式都具有这样的形式: $X_{i-1}\sqsubseteq X_i$,它是使用了 $k_i$ 次相同或不同的 $\Gamma_1\cup\Gamma_2$ 中的包含式: $\{\delta_1^i\sqsubseteq\sigma_1^i;\delta_2^i\sqsubseteq\sigma_2^i;\ldots;\delta_{k_i}^i\sqsubseteq\sigma_{k_i}^i\}$ 通过多次使用 $SL-\{\rightarrow\text{-rule}\}$ 的规则得到的,可以看作框架 $C_i[\ ]$ 使得 $X_{i-1}\equiv C_i[\ldots(\delta_1^i)\ldots(\delta_{k_i}^i)\ldots]\sqsubseteq C_i[\ldots(\sigma_1^i)\ldots(\sigma_{k_i}^i)\ldots]\equiv X_i$.

注意, $C_1,C_k$ 是红色的,且 $(X_k\sqsubseteq B)\equiv(C_k[\ldots(\delta_1^k)\ldots(\delta_{k_k}^k)\ldots]\sqsubseteq C_k[\ldots(\sigma_1^k)\ldots(\sigma_{k_k}^k)\ldots])$.

**断言 3**. 由每一个 $\delta_i^k|\sigma_i^k$ 出发向左搜索,一定能找到路径(可能包含多个并行路径),在子公式连接的意义下,

一直联通到若干个 $\left\{\delta_{i_1}^1\middle|\sigma_{i_1}^1,\dots,\delta_{i_m}^1\middle|\sigma_{i_m}^1\right\}$,且该路径的解读是 $\equiv X_0 \sqsubseteq X_1 \sqsubseteq X_2 \sqsubseteq \dots \sqsubseteq X_k \sqsubseteq X_{k+1} \equiv B$ 的一个子包含链.

证明:不失一般性,假定 $\delta_1^k$ 是灰色的.由 εL 公式的构成可知, $\delta_1^k \equiv \exists R.C$, 或者 $\delta_1^k \equiv A \sqcap B$.

首先考虑 $\delta_1^k \equiv \exists R.C$ 的情况.由有效链的构成,必有 $\left(C_{k-1}\left[\dots\left(\sigma_1^{k-1}\right)\dots\left(\sigma_{k_i}^{k-1}\right)\dots\right]\right) \equiv \left(C_k\left[\dots\left(\delta_1^{k-1}\right)\dots\left(\delta_{k_k}^{k}\right)\dots\right]\right)$. 由于 $\exists R.C$ 在右侧公式出现,它必然也在左侧公式中出现,它们只是同一个公式 $X_k$ 的两种不同的框架写法.注意到 $\delta_1^k \equiv \exists R.C$ 作为子公式,它在左式 $\left(C_{k-1}\left[\dots\left(\sigma_1^{k-1}\right)\dots\left(\sigma_{k_i}^{k-1}\right)\dots\right]\right)$ 中的位置有以下 4 种情形.

1. 是某 $\sigma_j^{k-1}$ 的子公式.此时有 $\delta_j^{k-1}\middle|\sigma_j^{k-1} \xleftarrow{\gamma.C} \delta_i^k\middle|\sigma_i^k$.

2. 等于某 $\sigma_j^{k-1}$.于是可得 $\delta_j^{k-1}\middle|\sigma_j^{k-1} \leftrightarrow \delta_i^k\middle|\sigma_i^k$.

3. 真覆盖某 $\sigma_j^{k-1}$.于是有 $\delta_j^{k-1}\middle|\sigma_j^{k-1} \xleftarrow{\gamma.C} \delta_i^k\middle|\sigma_i^k$.

以上 3 种情形皆可由出发向左搜索子公式连接.

4. $\delta_1^k$ 不与任何 $\sigma_j^{k-1}$ 相交.因为 $\exists R.C$ 是不可拆的,即它实际上是 $(\exists R.C)$, 从而不与任一 $\sigma_j^{k-1}$ 有公共部分,因此它完全是在框架 $C_{k-1}$ 中出现.但这在 $C_{k-1}$ 中不可能发生,因为 $C_k$ 是红色的. $C_{k-1}$ 是灰色的只在 $\sigma_j^{k-1} = \exists R.C[\ ]$ 是黑色时才发生.那么路径一定走到 $\sigma_j^{k-1}$.若它仍然是 $\exists R.C$ 型的,它在 $C_{k-2}$ 也就在框架中了;否则,(向左)传递下去很可能还在下一个框架中.但由于 $C_1$ 是红色的, $\sigma_j^{k-1}$ 一定会在之前与某 $\sigma_i^m$ 有子公式连接,这样一直延续到最左端,或者 $\sigma_i^m$ 成 $A \sqcap B$ 型.采用简单的归纳推理即可证明这一断言.

再考虑 $\delta_1^k \equiv A \sqcap B$ 的情形.此时, $A$ 可能落到 $\sigma_j^{k-1}$ 中,而 $\sqcap B$ 却在框架 $C_{k-1}$ 中.若先看 $A$ ,或者 $A = \delta_j^{k-1}$, 形成分支 $\delta_j^{k-1}\middle|\sigma_j^{k-1} \leftrightarrow A$. 若 $A$ 只是 $\sigma_j^{k-1}$ 的一部分,注意到 $\sigma_j^{k-1}$ 此时必须是 $D \sqsubseteq A$ 的形式,而 $\delta_j^{k-1} \sqsubseteq A$ 也是 $\Gamma_2$ 的包含式,子公式连接继续.至于 $\sqcap B$, 由于它是灰色的,属于框架,则它在 $C_1[\ ]$ 之前一定"消失",即落在某一个 $\sigma_i^m$ 中,子公式连接成功.正如例 5 的形式.

同样采用归纳法,可证明断言是成立的. □

3.3.2 $\langle \Gamma_2; \xrightarrow{\gamma.C} \rangle$ 中路径隐含信息的解读

将任意包含式 $A \sqsubseteq B$ 简写为 $A|B$. $A_1|B_1 \sim A_2|B_2 \sim \dots \sim A_k|B_k$ 是 $\langle \Gamma_2; \xrightarrow{\gamma.C} \rangle$ 中的路径或复合路径,其中, $\sim \in \{\xrightarrow{\gamma.C}; \xleftarrow{\gamma.C}; \leftrightarrow\}$.

1) $A_i|B_i \xrightarrow{\gamma.C} A_{i+1}|B_{i+1}$. 此时, $A_{i+1} \equiv C_\gamma[B_i]$, 唯一对应的包含链是 $C_\gamma[A_i] \sqsubseteq C_\gamma[B_i] \equiv A_{i+1} \sqsubseteq B_{i+1}$.

2) $A_i|B_i \xleftarrow{\gamma.C} A_{i+1}|B_{i+1}$. 与 1)类似,唯一对应的包含链是 $A_i \sqsubseteq B_i \equiv C_\gamma[A_{i+1}] \sqsubseteq C_\gamma[B_{i+1}]$.

3) $A_i|B_i \xrightarrow{\gamma.C_1} A_{i+1}|B_{i+1} \xrightarrow{\gamma.C_2} A_{i+2}|B_{i+2}$. 可产生的包含链是 $C_2[C_1[A_i]] \sqsubseteq C_2[C_1[B_i]] \equiv C_2[A_{i+1}] \sqsubseteq C_2[B_{i+1}] = A_{i+2} \sqsubseteq B_{i+2}$.

4) $A_i|B_i \xleftarrow{\gamma.C_1} A_{i+1}|B_{i+1} \xleftarrow{\gamma.C_2} A_{i+2}|B_{i+2}$. 与情形 3)正好对称.

5) $A_i|B_i \xrightarrow{\gamma.C_1} A_{i+1}|B_{i+1} \xleftarrow{\gamma.C_2} A_{i+2}|B_{i+2}$. 产生的包含链是 $C_1[A_i] \sqsubseteq C_1[B_i] \equiv A_{i+1} \sqsubseteq B_{i+1} \equiv C_2[A_{i+2}] \sqsubseteq C_2[B_{i+2}]$.

6) $A_i|B_i \xleftarrow{\gamma.C_1} A_{i+1}|B_{i+1} \xrightarrow{\gamma.C_2} A_{i+2}|B_{i+2}$. 此时,除非 $C_1[A_i]$ 仍是 $A_{i+2}$ 的子公式,即 $C_2[\ ] = C_1[C_1[\ ]]$, 才能有 $C_1[A_i] \subseteq C_1[B_i] \equiv C_2[A_{i+1}] \subseteq C_2[B_{i+1}] \equiv A_{i+2} \subseteq B_{i+2}$, 否则不可能连接成包含链.

7) 如果 $A|B \to D|E$, $B$ 在 $E$ 中有多个出现, $D = C[B \dots B \dots]$, 约定其解读为 $C[A \dots B \dots] \sqsubseteq C[B \dots B \dots] \equiv D \sqsubseteq E$.

由以上的基本细节分析可知:若 $A_1|B_1 \sim A_2|B_2 \sim \dots A_k|B_k$ 中的初等包含式 $\{A_i \sqsubseteq B_i | 1 \leqslant i \leqslant k\}$ 是语法上两两不相同的,同时对它的解读可形成一个标准包含链(即每一基本单元都是基本的),则解读出的包含链是唯一的.但如果其中出现两个相同的基本包含式,或更多相同的基本包含式,则该路径在可解读的前提下,产生无穷多个标准包含链.一个简单的例子是 $A_1|B_1 \xrightarrow{\gamma.C} A_1|B_1$ 可以任意延长: $A_1|B_1 \xrightarrow{\gamma.C} A_1|B_1 \xrightarrow{\gamma.C} A_1|B_1 \dots \xrightarrow{\gamma.C} A_1|B_1$, 也可

能存在多种嵌套循环的路径.

**定义 10(有效路径)**. 如果 $\langle \Gamma_2; \xrightarrow{\gamma.C} \rangle$ 的路径产生了有效链,则称其为有效路径,称该链为原始有效链.该有效链产生的结论 $X \sqsubseteq Y$ ($X,Y$ 分别是链的最左端公式和最右端公式),称为待检测包含式.显然, $Sig(X \sqsubseteq Y) \sqsubseteq Sig(T_i)$, $\Delta$ 中的包含式都是待检测包含式.

由定义可知,待检测包含式都是图 $T$ 中的路径产生的.

任给一个由 $SL + \Gamma_2$ 推出的有效链 $\Phi: A \sqsubseteq X_1 \sqsubseteq X_2 \sqsubseteq ... \sqsubseteq X_k \sqsubseteq B$, 其中, $Sig(A) \bigcup Sig(B) \subseteq Sig(T_1), Sig(X_1) \bigcap \left(Sig(T_2) - Sig(T_1)\right) \neq \varnothing, \forall i, 1 \leqslant i \leqslant k$.同时,都是灰色的公式,且每一个包含式都是基本包含式.给定红色框架 $C[\ ]$,它有 1 个或多个变元位置,我们将给定有效链的每一个公式都分别植入框架的空位置中,得到另一个有效链,该链的每一包含式仍是基本包含式.即

$$C[\Phi]: C[A] \subseteq C[X_1] \subseteq C[X_2] \subseteq ... \subseteq C[X_k] \subseteq C[B].$$

**定义 11(正规有效链)**. $\Phi$ 是正规有效链,当且仅当不存在另一有效链 $\Phi_1$ 使得 $\Phi \equiv C[\Phi_1]$, 其中, $C[\ ]$ 是一个框架.

**引理 8(归约引理)**.

1) $\langle \mathcal{F}; \leqslant \rangle$ 中的任一有效链 $A \rightsquigarrow B$ 都等价于一条正规有效链.

2) 任一正规有效链是一条或多条原始有效链通过若干次框架操作而成的.

证明:1)的证明是显然的.2)的证明可使用归纳法.$\Delta$中的平凡有效链自身就是原始的.非平凡的最小情形是长度为 2 的正规有效链 $A \sqsubseteq B \sqsubseteq C$. 该链由两个基本包含式组成: $A \sqsubseteq B, B \sqsubseteq C$. 若二式都在 $\Gamma_2$ 中,则引理得证.否则,其中某一公式的长度一定大于给定的整数 $p$.下面分情形讨论.

(1) 若 $A \sqsubseteq B$ 不在 $\Gamma_2$ 中,则 $A \sqsubseteq B$ 按基本包含式的定义,必定是一次或多次执行 $\exists^+, \sqcap^+, \sqcap^-, \sqcap_+$ 的结果.若 $A$ 最外面的框架形如 $\exists R.[\ ]$, $A \equiv \exists R.[A_1], B \equiv \exists R.[B_1]$, 这里最外面的框架也就是最后执行的框架操作而得的.注意有 $A_1 \sqsubseteq B_1$. 此时, $B \sqsubseteq C$ 必须是在 $\Gamma_2$ 中.否则, $B \sqsubseteq C \equiv \exists R.[B_1] \sqsubseteq \exists R.[C_1]$, 因为这是同样的一个框架操作 $\exists R.[\ ]$ 而得到的,同时 $B_1 \sqsubseteq C_1$ 仍然成立.那么, $A_1 \sqsubseteq B_1 \sqsubseteq C_1$ 仍然成立.这与 $A \sqsubseteq B \sqsubseteq C$ 是正规有效链的假设矛盾.所以,在此情形下可得到 $\langle \Gamma_2; \xrightarrow{\gamma.C} \rangle$ 中的有效路径 $A|B \xrightarrow{\gamma.\exists R.[\ ]} B|C$.

(2) 若最后的操作是 $\sqcap_+$, $A \equiv A_1 \sqcap A_2$.而 $A_2 \sqsubseteq B$ 仍是基本包含式.此时若仍然有 $p < |A_2| < |A|$, 检查第 2 个基本包含式,判定其是否由 $\sqcap_+$ 得到.若是,则该过程必在某一基本包含式 $A_k \sqsubseteq B$ 停止.若它在 $\Gamma_2$ 中,则考虑 $B \sqsubseteq C$,若该包含式也在 $\Gamma_2$ 里,则路径 $A_k|B \leftrightarrow B|C$ 即是要求的路径,因为 $A \sqsubseteq C$ 是 $A_k \sqsubseteq C$ 在 $T_1$ 中的自然逻辑结论.若 $B \sqsubseteq C$ 不在 $\Gamma_2$ 中,则有框架 $C[(\ )_1, ..., (\ )_m, ...]$, 使 $B \equiv C[(B_1)_1, ..., (B_m)_m, ...], C \equiv C[(C_1)_1, ..., (C_m)_m, ...]$, 其中, $B_1 \sqsubseteq C_1, ..., B_m \sqsubseteq C_m$ 都是 $\Gamma_2$ 的基本包含式.对应的复合路径是 $\{A_k|B \leftarrow B_i|C_i|1 \leqslant i \leqslant m\}$.

(3) 如果 $A_k \sqsubseteq B$ 仍然不在 $\Gamma_2$ 中,则框架只能是 $\exists R.[\ ]$ 或者 $A_k \equiv (A^1 \sqcap A^2), B \equiv (B^1 \sqcap B^2)$. 前者在情形(1)中已经证明了.现在考虑后者,注意到在 $\sqcap^+$ 的操作中,前提条件是: $A^1 \sqsubseteq B^1; A^2 \sqsubseteq B^2$ 仍是基本包含式.考虑: $(B \sqsubseteq C) \equiv (B^1 \sqsubseteq B^2) \sqsubseteq C$. 如果此式在 $\Gamma_2$ 中,则复合路径 $A^i|B^i \leftrightarrow B^1|C, i \in \{1,2\}$, 可得 $A_k \sqsubseteq C$.若此式不在 $\Gamma_2$ 中,则框架只能由 $\sqcap_+, \sqcap^+$ 中产生.在 $\sqcap_+$ 的情形,比如, $B^2 \sqsubseteq C$ 是基本包含式,那么 $A^2 \sqsubseteq B^2 \sqsubseteq C$ 是有效的,其复杂度削减了,根据归纳假设,它必定内含复合路径,这决定了 $A^2 \sqsubseteq C$, 而 $A \sqsubseteq C$ 仍是 $A^2 \sqsubseteq C$ 在 $\Gamma_2$ 的自然结论.在 $\sqcap^+$ 的情形, $C \equiv C_1 \sqcap C_2$, 同时, $B^1 \sqsubseteq C_1, B^2 \sqsubseteq C_2$ 仍然是基本包含式,于是, $A_k \sqsubseteq B^1 \sqsubseteq C_1$ 被拆分为 $A^1 \sqsubseteq B^1 \sqsubseteq C_1, A^2 \sqsubseteq B^2 \sqsubseteq C_2$. 根据归纳假设,二者分别为有效复合路径 $\eta_1, \eta_2$ 所表达.显然,有效复合路径 $\eta_1 + \eta_2$ 表达了 $A_k \sqsubseteq B \sqsubseteq C$. 从而自然地由 $T_1$ 推得 $A \sqsubseteq C$. □

以上陈述了归纳证明的基础部分,只是给出逻辑思路有一个直观体验.注意到,若 $\{A \sqsubseteq C; B \sqsubseteq C\}$ 中有一个包含式在 $\Gamma_2$ 中,则结论的正确性是很好理解的.为了完成归纳过程,需要以下的断言.

**断言 4**. 任给有效链,如果其中存在 $\Gamma_2$ 的包含式,则该有效链必定是 $\langle \Gamma_2; \xrightarrow{\gamma.C} \rangle$ 中某个复合路径的解读.

证明的思路可仿照前述断言 3 的过程进行,此处从略.

下面给出断言 3 的一般情形的归纳证明.

证明:令 $A\sqsubseteq X_1\sqsubseteq X_2\sqsubseteq...\sqsubseteq X_k\sqsubseteq B$ 是正规有效链.考虑最右边的包含式 $X_k\sqsubseteq B$:若它本身就在 $\Gamma_2$ 中,由断言,结论成立.若不是,它必须由框架操作得到:注意这里 $k>2$,$\sqcap_+$ 是不能在包含式左端施行的.否则该链不是正规的.只需考虑框架操作 $\sqcap^+$, $\exists R.[\ ]$ (若它是 $\sqcap_-$ 的结果,在此没有影响,因为该操作不产生框架)

若最后的操作是 $\exists R.[\ ]$,则它可能一直是向左每一公式的最外面的部分,也可能消失,但它绝对不可能是每一包含式的外来框架,否则该有效链不是正规的.于是向左逐个检测,一定有 $i,1\leqslant i\leqslant k$;$X_1$ 的最外部分是 $\exists R.[\ ]$ 而 $X_{i-1}\sqsubseteq X_i$ 必须在 $\Gamma_2$ 中.由断言,该正规有效链是 $\langle\Gamma_2;\xrightarrow{\gamma.C}\rangle$ 中某个复合路径的解读.

若最后的操作是 $\sqcap^+$, 则 $B\equiv B_1\sqcap B_2,X_k\equiv X_k^1\sqcap X_k^2$, 同时, $X_k^1\sqsubseteq B_1,X_k^2\sqsubseteq B_2$ 是基本包含式.于是,原正规有效链被拆分成两个复杂度低的有效链(使用 $\sqcap^-$ ):

$$A\sqsubseteq X_1\sqsubseteq X_2\sqsubseteq...\sqsubseteq X_k^1\sqsubseteq B,$$
$$A\sqsubseteq X_1\sqsubseteq X_2\sqsubseteq...\sqsubseteq X_k^2\sqsubseteq B.$$

根据归纳假设,它们分别由有效路径 $\eta_1,\eta_2$ 解读,容易验证,原正规有效链必定是 $\eta_1+\eta_2$ 的解读. □

**定理 2(归约定理)**. $|T_1\cup T_2|$ 是 $T_1$ 的保守扩充当且仅当 $T_1\vdash X\sqsubseteq Y$, 其中 $X\sqsubseteq Y$ 遍历 $\Gamma_2$ 的所有待检测包含式(含$\Delta$).

由引理 8,归约定理是显然成立的(参见定义 10).记所有待检测包含式构成的集合为$\Omega$.注意$\Omega$可能是无穷的,要检测保守扩充性,必须有如下保证.

**推论 1**. 定义 $\Omega_0=\{X\sqsubseteq Y|(X\sqsubseteq Y)\in\Omega\}_1$, 其中,$X,Y$ 的出度都不大于 $|T_1\cup T_2|$. 那么, $|T_1\cup T_2|$ 是 $T_1$ 的保守扩充当且仅当 $\Omega_0$ 的任一包含式都是 $T_1$ 的逻辑结论.

证明过程见文献[11].注意到给定待检测包含式 $X\sqsubseteq Y$, 判定 $T_1\vdash X\sqsubseteq Y$ 是否成立的经典模型的模拟算法复杂性是多项式的.引起计算复杂性急剧上升到指数阶的原因是,必须搜索 $\Omega_0$ 的所有成员.为简化这一繁琐的工作,接下来将细化并适当扩展 $\Gamma_2$, 使所有 $\Omega_0$ 的成员信息都在一个图中清晰地显示出来.在构成这样的一个图后,验证保守扩充的工作的复杂度就是多项式的了.下一节将给出的基于 $\Gamma_2$ 建立的并发进程图就能达到此目的.

### 3.4 并发进程图 $\Upsilon$ 构建

在给出核心构建前,需要以下准则.

**命题 2(无效的“子公式-框架连接”判定准则)**.

1) 如果在图 $\Gamma_2$ 的一个连接路径中出现形如 $A_1|B_1\xrightarrow{\beta.C}A_2|B_2$ 的链接,其中 $C[]$ 是 $\exists R.[]$ 型的, $R$ 为黑色.这就意味着 $C[B_1]\equiv A_2$, $C[]$ 是灰色的.在向左解读时,每一个公式有可能都要加上框架 $C[]$,直至路径的最左件(记为 $E|A'$), 那么 $C[E]$ 不是红色的,路径不能解读为有效路径.除非向左 $p$ 步之前,路径中有 $\Gamma_2$ 的包含式 $A^i|B^i,|B^i$ 将消化掉 $C[A_1]$. 即,若由 $C[A_1]$ 向左到 $B^i$ 生成的框架为 $C[...[A_1]...]$, 只有在后者是前者的子公式的情况下,我们才可能解读到一个有效链.

2) 若 $A_1|B_1\xleftarrow{\beta.C}A_2|B_2$ 出现在路径中,则其(有)无效性正好与上述情况对称.

3) $A_1|B_1\xleftarrow{\beta.C}A_2|B_2\xrightarrow{\beta.C}A_3|B_3$ 的情形:记 $B_1=C_1[A_2],A_3=C_2[B_2]$, 此时,当且仅当 $C_1[]\neq C_2[]$ 时,路径解读不能继续.注意,即使可以继续解读,该链接也不对路径两端的红色公式产生影响.

4) $A_1|B_1\xleftarrow{\alpha.C}A_2|B_2\xrightarrow{\alpha.C}A_3|B_3$ 的情形和情形 3)一样.

5) 最重要的一个细节是,若 $A_1|B_1\xrightarrow{\beta.C}A_2|B_2$, 而 $A_2=D_1\sqcap D_2,A_2|B_1\xrightarrow{\alpha.C}D_1,D_1$ 和 $D_2$ 同时是灰色的.此时,如果 $\Gamma_2$ 中有一形如 $X\sqsubseteq D_2$ 的链接,则保留该链接,但在 $D_2$ 处做一个待定处理的标记 $D_2(?)$. 按情形 1),如果能够通过,则记为 $D_2(!?)$; 若不能通过,则记为 $D_2(x?)$.

6) 在情形 5)中,如果 $D_2$ 是红色的,则可通过.若 $\Gamma_2$ 中有形如 $X\sqsubseteq D_2$ 的包含式,则仍然加上标记(?).

3.4.1 并发进程图 $\Upsilon$

以希腊小写字母表示包含式.令 $\langle\zeta_{11},\zeta_{21},\ldots,\zeta_{k1}\rangle,\langle\theta_{1m},\theta_{2m},\ldots,\theta_{nm}\rangle$ 表示 $\Gamma_2$ 的最左、最右的竖直向量.拟建立 $\Upsilon$ 的由 $k$ 个子图组成,每一子图 $\Upsilon_i$ 是都是拟树形的,以 $\zeta_{1i}$ 为树根.设 $\zeta_{1i}\equiv E_{1i}\sqsubseteq X_{1i}$,$\Upsilon_i$ 的目的是清晰显示由 $\zeta_{1i}$ 到达每一个 $\theta_{jm}$ 的有效路径.

记 $\zeta_{11}$ 为 $E_1\sqsubseteq X_1$.建立拟树形的图 $\Upsilon_1$ 如下:

(1) $\Upsilon_1^0=E_1\sqsubseteq X_1$.

(2) $\Upsilon_1^1=\Upsilon_1^0\cup\left\{X_1\sim Y_1^1;X_1\sim Y_1^2;\ldots;X_1\sim Y_1^{k_1}\right\}$,其中,$\sim\in\left\{\xrightarrow{\gamma.C};\xleftarrow{\gamma.C};\leftrightarrow\right\}$,$Y_1^i\sqsubseteq X_2^i$ 是 $\Gamma_2$ 的包含式.

(3) 树形图的每一个节点是 $\Gamma_2$ 的包含式.如果同一包含式的左、右件也有子公式连接关系,则明确地标示出来.

(4) 若 $X_i^1,X_i^2,\ldots,X_i^{k_i}$ 是 $\Upsilon_1^i$ 的右边公式($i$ 层包含式的右件),则 $\Upsilon_1^{i+1}=\Upsilon_1^i\cup\left\{X_i^j\sim Y_i^j\right\}$.

(5) 若 $E_1$ 与某一 $\theta_{im}=X_j\left|F_i\right.$ 可以连接上,则一定有这样的链接 $E_1\left|X_1\sim Y_1\right|X_2\sim\ldots\sim X_j\left|E_i\right.$,该链中出现的包含式两两不等.

(6) 每扩张一次 $Y_1^i$,按命题 2 的准则对树枝进行修剪(加 (!?),($x$?) 符号);或剪枝,即终止无效的链接.

显然,$T_2$ 中包含式的个数远小于 $\left|T_2\right|$.由 $\Gamma_2$ 的构建可知,$\Gamma_2$ 的包含式个数也很小.实际上可以只使用 $\sqcap\sqsubset$ 规则来构成 $\Gamma_2$ 同时不会破坏 $\Gamma_2$ 中路径的饱和性.以下以 $m$ 记 $\Gamma_2$ 的包含式个数.由于有效循环式可能存在,每一 $\Upsilon_1^i$ 仍然可扩张下去.但是 $\Gamma_2$ 的包含式形成的子集个数是 $2^m$,那么在 $2^m+1$ 步时或之前,一定有 $i<j$,使得 $\Upsilon_1^i$ 和 $Y_1^j$ 最右边的包含式集合相等(若某包含式多次出现,只算 1 个).将这个集合记为 $\Phi=\left\{X_1\left|Y_1,X_2\right|Y_2,\ldots,X_n\left|Y_n\right.\right\}$,其中的包含式两两不等.

注意到,凡是与 $E_i$ 能链接上的 $F_j$ 及其各种非循环连接都已经在 $\Upsilon_1^i$ 之前出现过了.不抹掉任何 $\Upsilon_1^i$ 的连接,分别将 $\Upsilon_1^i$ 和 $\Upsilon_1^j$ 中的这一集合,凡是出现多次的同一包含式(多个节点)都收缩成一个包含式(一个节点)而连接自然顺着节点的合并,自然地压缩在 $\Phi_1$ 和 $\Phi_2$ 之间:这里 $\Phi_1$ 和 $\Phi_2$ 是同一个集合,是两个竖直的竖列,它们分别处于并发进程的 $i$ 位置和 $j$ 位置.以 $\Psi\equiv\langle\Phi_1;\Phi_2;\sim\rangle,\sim\in\{\xrightarrow{\gamma.C};\xleftarrow{\gamma.C};\leftrightarrow\}$ 记这个连接图.注意到 $\Upsilon_1^{j+j-i}$ 是集合 $\Phi$ 在其右边包含式的第 2 次出现.本来是发散的树形被压缩成 $\Upsilon_1^i\left|\Psi\right|\Psi\ldots$ 的形式.我们或许可将循环部分增加到 $\left|T_1\cup T_2\right|$ 多个单位.单一重复的操作复杂度是 $O(\left|T_1\cup T_2\right|)$ 的,但它不是最佳的选择:在搜索的优化算法 $\Omega_0$ 中,$\left\{\left|T_1\right|,\left|T_2\right|\right\}$ 处于此消彼长的辩证关系.这个界不可能是绝对的,其中有很大的优化空间.

**断言 5**.

1. 若某个 $F_i$ 不在 $\Upsilon_1^i$ 出现,则不存在 $E_1,F_i$ 的有效连接,且 $F_i$ 也不会在 $\Phi$ 中出现.

2. 若 $F_i$ 在 $\Upsilon_1^i\,|\,\Phi$ 中出现,则任何由 $\Phi$ 中的 $F_i$ 到 $E_1$ 的路径,其节点没有标记 ($x$?),一定是有效路径.若所有的有效路径都不加宽 $E_1$ 和 $F_i$,则 $\Omega_0$ 的搜索在 $\Upsilon_1^i$ 终止.如果有增量,令 $t$ 是最小的增量,又令 $n=\dfrac{\left|T_1\right|}{t}$,那么 $\Omega_0$ 的搜索将在图 $\Upsilon_1^i\left|\Phi\right|\ldots\left|\Phi\right|$ 中终止.为了方便,将此图表示为 $\Upsilon_1^i\left|\Phi^n\right.$.

3. 事实上,$\Upsilon_1^i\left|\Phi\right|\Phi\left|\ldots\right|\Phi\right|$ 的子图 $\Upsilon_1^i\left|\Phi\right|$ 已经拥有足够的信息量,通过如下步骤,可以容易地将 $E_1$ 和任一 $F_j$ 的所有路径及所有可能的循环都找出来:

(1) 连接 $E_1$ 和 $F_j$ 所有无循环路径.

(2) 如果有循环路径,则该循环一定是捆绑在无循环路径的某个节点上,并且每一个循环对 $E_1$ 和 $F_j$ 的框架扩展都有明确的标号信息.

(3) 给出一个连接 $E_1$ 和 $F_j$ 所有无循环路径,并给出在每一个节点上可能的循环及循环对 $E_1,F_j$ 框架扩展的信息,在该路径上任意执行各种循环的组合,如循环 $\rho_1$ 执行 2 次,同时循环 $\rho_2$ 执行 1 次.这样就可以直接得到所有 $\Omega_0$ 的元素.

**定义 12**(图 $\Upsilon$ 的构建).

$$\Upsilon \equiv \begin{cases} \gamma_1^{i_1} \mid \Phi^{n_1} \\ \gamma_2^{i_2} \mid \Phi^{n_2} \\ \gamma_3^{i_3} \mid \Phi^{n_3} \\ \ldots \\ \gamma_k^{i_k} \mid \Phi^{n_k} \end{cases}.$$

3.4.2 $\Omega_0$ 公式的获得

1) 任一由某个 $E_i$ 出发向右达到某一 $F_j$ 的路径通过解读得到的包含链,若 $A,B$ 分别是最左端和最右端公式(红色),则 $A \sqsubseteq B$ 是 $\Omega_0$ 的成员(出度≤$|T_1 \cup T_2|$).

2) 路径的?型节点的归纳处理:$\varepsilon$ 是 $\theta_j : F_j |X_j$ 到 $\zeta_i : X_n |E_i$ 的路径.路径中第 1 次出现 $D_1 \sqcap D_2(?)$.如果 $\Upsilon$ 不出现形如 $A|D_1 \sqcap D_2 \leftarrow D_2|B$ 的连接,标记若是 $(x?)$,则可读下去.若标记是 $(x?)$,则放弃.

3) 若出现形如 $A|D_1 \sqcap D_2 \leftarrow D_2|B$,解读 $D_2|B$ 向左的路径在 $D_2$ 不加上任何框架的条件下,同时 $D_1$ 向左的路径解读也不会加上框架,则两条路径复合可读,具体为 $E^1 \sqsubseteq \ldots \sqsubseteq D_2; E^2 \sqsubseteq \ldots \sqsubseteq D_1; C[D_1 \sqcap D_2] \sqsubseteq (F_j)$,则 $C[E^1 \sqcap E^2] \sqsubseteq \ldots \sqsubseteq C[D_1 \sqcap D_2] \sqsubseteq \ldots \sqsubseteq (F_j)$,其中,$(F_j)$ 是前段解读的结果,$F_j$ 被套入了一些必要的框架.

4) 由情形 2),继续处理前面两个包含式链中 $\sqcap$ 中的(?)公式.

5) 至于 $T_2$ 中的包含式左件是多个公式的 $\sqcap$ 的情形,类似处理即可.

**断言 6**. 对 $\Upsilon$ 的路径解读,可得到所有 $\Omega_0$ 的公式.

这是因为 $\Gamma_2$ 路径的饱和性及每一个 $\Upsilon_j^{i_j} \mid \Phi^{n_j}$ 的构成,它保证了形为 $\exists R.[\ ]$ 的嵌套达到了 $|T_1 \cup T_2|$.

### 3.5 算法和复杂度分析

接下来对 $\Omega_0$ 的包含式进行检测,当然每一次检测是多项式复杂度的.为避免无效的工作,按照从易到难的原则进行.

1) 算法分析

(1) 对 $\Upsilon_1^{i_1}, \Upsilon_2^{i_2}, \ldots, \Upsilon_k^{i_k}$ 的每一个包含式 $A \sqsubseteq B$ 进行经典模型 $\langle T_1; \sqsubseteq \rangle$ 检测.如果遇到 $T_1 \notin A \sqsubseteq B$,则停机.输出"no".

(2) 按顺序检测 $\Upsilon_1^{i_1}|\Phi, \Upsilon_2^{i_2}|\Phi, \ldots, \Upsilon_k^{i_k}|\Phi$ 的每一个包含式 $A \sqsubseteq B$.

(3) 按照上式,逐个延长图一个 $\Phi$ 单位.

(4) 若 $\Upsilon$ 产生的包含式都检查完毕,则 $|T_1 \cup T_2|$ 是 $T_1$ 的保守扩充.

2) 复杂度分析

(1) 构造的 $\Gamma_2$ 复杂度不超过 $2^p$.这里,$p$ 是 $T_2$ 中最长公式的长度.

(2) 构造 $\Upsilon$ 的复杂度不超过 $m \times 2^k$,这里 $k$ 是 $T_2$ 的包含式个数,且 $m < k$,$m$ 是 $T_2$ 中有效左件的个数,是包含式集合的一个子集.

(3) 路径长度 $\sigma < 2^k$.路径在 $\Phi$ 出现之后就开始出现同步循环,所以满足饱和性所要求的长度不会很长,路径的个数上界不超过 $\sigma^k$.

(4) $2^p + m \times 2^k + \sigma^k$,尽管它仍是指数形式,但它远远小于 $|T_1 \cup T_2|$,后者是 $T_1 \cup T_2$ 中所有符号出现的次数之和.

## 4 总结及后续研究

本研究领域中当前公认的保守扩充检测的算法要对几乎 $2^{|T_1 \cup T_2|}$ 个经典模型进行测试,其中被测试的主要公式 $C, Sig(C) \subseteq Sig(T_1)$,而 $C$ 中 $\exists R.$ 型的嵌套深度的上界必须是 $|T_1 \cup T_2|$,这是 $T_1, T_2$ 中所有公式的长度之和.本文试图构建一种图推理的办法来降低推理算法的复杂度.我们的方法是将所有可能的 $T_1, T_2$ 的推理信息都压缩在一个图 $\tau$ 中,该图的任意一个包含式的公式长度都不大于原来最长的公式.这个界比 $|T_1 \cup T_2|$ 小得多,更比 $C$ 的长

度小得多.特别是一旦 $\Upsilon$ 建立起来后,检测保守扩充的算法相对于 $\Upsilon$ 的路径个数就是多项式的.

本文的图推理作为一个本体库的基本推理设施 $\langle T;\xrightarrow{C}\rangle$ 有两个好处.一是它的构建比较简单;二是该设施对本体库的知识更新的维护、检测新的知识引入后本体是否受到损害,是一个很方便的工具,只需在图中添加若干新的包含式及子公式关系连接,对新的路径进行检测即可.

本文建立的推论机制具有通用性,对任何 DL-Lite 家族的系统,只要删除或添加一两条规则即可适用.

今后的研究工作主要是:能否从连接图 $\Gamma_1|\Delta|\Gamma_2$ 中发现红色连接及灰色连接的一些可形成同态模拟式的概念,它们可以直接决定保守扩充的关系?这样就不必逐一对 $\Omega_0$ 的包含式做形式模型检测.具体来说,注意到图的另一子图 $\langle \Gamma_1|\Delta|\Sigma;\xrightarrow{\alpha.C}\rangle$ 中的有效路径同样可能产生循环,能够模拟 $\langle \Gamma_2;\xrightarrow{\gamma.C}\rangle$ 中任意循环路径的行为,特别是这些行为对有效路径的最左公式和最右公式的框架操作结果是一样的.那么,所有的待测试包含定理是 $T_1$ 的逻辑结论,从而 $|T_1\cup T_2|$ 是 $T_1$ 的保守扩充;反之,若 $\langle \Gamma_2;\xrightarrow{\gamma.C}\rangle$ 中存在某个有效路径的循环或不循环的行为不能在 $\langle \Gamma_1|\Delta|\Sigma;\xrightarrow{\alpha.C}\rangle$ 中得到模拟,则 $|T_1\cup T_2|$ 不是 $T_1$ 的保守扩充.

两个子图的路径都是有穷的.判定保守扩充就等价于判定图的互模拟,而判定互模拟应该是多项式的.以空间换时间,判定算法的复杂度远低于 $2^{|T_1\cup T_2|}$.

注意到一条不循环路径在捆绑了所有可能的循环信息后,就可以解读为一个程序或一个可计算函数.这样,$\langle \Gamma_1|\Delta|\Sigma;\xrightarrow{\alpha.C}\rangle$ 和 $\langle \Gamma_2;\xrightarrow{\gamma.C}\rangle$ 将分别等价于两个可计算函数的有穷集合 $\Psi_1,\Psi_2$.保守扩充问题就转换成为 $\Psi_2$ 是否可图灵规约到 $\Psi_1$ 的问题,即 $\Psi_2$ 中每一可计算函数是否都可用 $\Psi_1$ 来计算.

**致谢** 在此,特别感谢桂林电子科技大学计算机软件创新团队的支持.

**References**:

[1] Ghilardi S, Lutz C, Wolter F. Did I damage my ontology? A case for conservative extensions in description logics. In: Proc. of the KR 2006. 2006. 187−197.

[2] Lutz C, Walther D, Wolter F. Conservative extensions in expressive description logics. In: Proc. of the IJCAI 2007. 2007. 453−458.

[3] Lutz C, Toman D, Wolter F. Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system. In: Boutilier C, ed., Proc. of the 21st Int'l Joint Conf. on Artificial Intelligence (IJCAI 2009). 2009. 2070−2075.

[4] Li P. Modular construction and reuse of domain ontology [MS. Thesis]. Guilin: Guangxi Normal University, 2010 (in Chinese with English abstract).

[5] Shen YM, Wang J. Complexity of conservative extensions and inseparability in the description logic $\mathcal{EL}$. In: Zhao DY, Du JF, Wang HF, Wang P, Ji DH, Jeff ZP, eds. The Semantic Web and Web Science. CCIS 480, Heidelberg: Springer-Verlag, 2014. 78−86. [doi: 10.1007/978-3-662-45495-4_7]

[6] Nie DG, Kang WQ, Cao FS, Wang J. Containing reasoning and its conservative extensions in description logic $\mathcal{FL}_0$. Journal of Computer Research and Development, 2015,52(1):221−228 (in Chinese with English abstract).

[7] Baader F, Calvvanese D, Mcguinness DL, Nardi D, Patel-Schneider PF. The Description Logic Handbook: Theory, Implementation and Applications. Cambridge: Cambridge University Press, 2003.

[8] Lutz C, Wolter F. Conservative extensions in the lightweight description logic $\mathcal{EL}$. In: Pfenning F, ed. Proc. of the CADE 2007. LNCS 4603, Berlin, Heidelberg: Springer-Verlag, 2007. 84−99. [doi: 10.1007/978-3-540-73595- 3_7]

[9] Cuenca Grau B, Horrocks I, Kazakov Y, Sattler U. A logical framework for modularity of ontologies. In: Proc. of the Int'l Joint Conf. on Artificial Intelligence (IJCAI). AAAI, 2007. 298−304.

[10] Lutz C, Seylan I, Wolter F. An automata-theoretic approach to uniform interpolation and approximation in the description logic $\mathcal{EL}$. Journal of Internet Services & Applications, 2012,2(2):171−185.

[11] Lutz C, Wolter F. Foundations for uniform interpolation and forgetting in expressive description logics. In: Proc. of the IJCAI. 2011. 989−995.

[12] Wang Z, Wang K, Topor RW, Pan JZ. Forgetting for knowledgebases in DL-Lite. Annals of Mathematics and Artificial Intelligence, 2010,58(1-2):117–151. [doi: 10.1007/s10472-010-9187-9]

[13] Grau BC, Horrocks I, Kazakov Y, Sattler U. Just the right amount: Extracting modules from ontologies. In: Proc. of the WWW 2007. 2007. 717–726. [doi: 10.1145/1242572.1242669]

[14] Lutz C, Wolter F. Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. Journal of Symbolic Computation, 2010,45(2):194–228. [doi: 10.1016/j.jsc.2008.10.007]


**附中文参考文献**:


[4] 李璞.领域本体的模块化构建与重用[硕士学位论文].桂林:广西师范大学,2010.

[6] 聂登国,康旺强,曹发生,王驹.描述逻辑$\mathcal{FL}_0$包含推理及其保守扩充.计算机研究与发展,2015,52(1):221–228.



**王驹**(1950－),男,广西博白人,博士,研究员,教授,博士生导师,CCF 专业会员,主要研究领域为数理逻辑,计算机理论,语义网络,描述逻辑.



**余泉**(1979－),男,副教授,主要研究领域为模态逻辑,描述逻辑,多智能主体认知规划.



**陈光喜**(1971－),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为定理机器证明,图像处理.